ROHDE&SCHWARZ

Operating manual

LOGIC GENERATOR IGA

344.0014.04

R 43 559 - 1

Digitized by the Internet Archive
in 2024

https://archive.org/details/manualsplus_00850

Table of Contents

1	Characteristics and Uses	1.1
2	Preparation for Use and Operating Instructions	2.1
2.1	Preparation for Use	2.1
2.1.1	Legends	2.1
2.1.2	Adjusting the Set to the Available AC Supply Voltage	2.7
2.1.3	Connecting the Data Drivers and the Clock Driver	2.7
2.1.4	Switching the Set on	2.7
2.2	Manual Operation	2.8
2.2.1	Direct Module	2.8
2.2.1.1	Introduction	2.8
2.2.1.2	Entry	2.11
2.2.1.2.1	Entry of Data Format	2.11
2.2.1.2.2	Entry of Data Lines	2.14
2.2.1.2.3	Insertion of Additional Data Lines	2.16
2.2.1.2.4	Correction of Data Entered Wrongly or by Mistake	2.17
2.2.1.2.5	Cancelling Data Lines	2.18
2.2.1.2.6	Duplication of Data Lines	2.19
2.2.1.2.7	Renumbering of Data Lines	2.20
2.2.1.2.8	Entering the Frequency	2.21
2.2.1.2.9	Entry of Pause	2.22
2.2.1.2.10	Tristate Mode	2.23
2.2.1.3	Data Output	2.25
2.2.1.3.1	Data Drivers and Clock Driver	2.25
2.2.1.3.2	Multi-clock Driver	2.25
2.2.1.3.3	Start, Stop of Data Output	2.26
2.2.1.3.4	Adjusting the DAV Signal	2.26
2.2.1.3.5	Single-stepping	2.27
2.2.1.3.6	Clock Modes	2.27
2.2.1.3.7	Output Modes	2.27
2.2.2	Modules Defined by a Name	2.28
2.2.2.1	Assigning a Module Name	2.28
2.2.2.2	Fetching an Existing Module	2.29
2.2.2.3	Menu of the Modules Stored in the IGA and Changing of a Module Name	2.30
2.2.2.4	Erasing of Modules	2.31

2.2.2.5 Duplication of Modules 2.32
2.2.3 Sequences 2.33
2.2.3.1 Assigning a Module Sequence Name 2.33
2.2.3.2 Fetching an Existing Module Sequence 2.34
2.2.3.3 Entering Module Names to Create a Module Sequence 2.35
2.2.3.4 Menu of the Module Sequences Stored in the IGA; Erasing Module Sequences 2.37
2.3 Cassette Drive (Option IGA-B2) 2.38
2.3.1 Tape Formatting 2.38
2.3.2 Storing Data on Tape 2.39
2.3.3 Loading Data from Tape 2.41
2.3.3.1 APPEND Instruction 2.41
2.3.3.2 SUBSTITUTE Instruction 2.42
2.3.4 Write Protection 2.42
2.3.5 Erasing the Complete Contents of the IGA 2.43
2.4 Controlling the IGA via the IEC Bus (Option B1) 2.44
2.4.1 Setting the Device Address and the Terminating Character .. 2.44
2.4.2 Interface Functions 2.46
2.4.3 Setting Instructions 2.46
2.4.4 Data Direct Instruction 2.48
2.4.5 Printer Output 2.49
2.4.6 Group of Addressed and Universal Instructions 2.52
2.4.6.1 Remote/Local 2.52
2.4.6.2 Device Clear 2.52
2.4.6.3 Device Trigger 2.52
2.4.6.4 Service Request, Status Byte, Serial Poll 2.54
2.5 Error Messages 2.56
2.6 List of Events for 3-wire Handshake Operation 2.58
2.7 List of Events for 2-wire Handshake Operation 2.60

ROHDE&SCHWARZ

IGA

LOGIC GENERATOR IGA

32 data channels, max. clock frequency 10 MHz

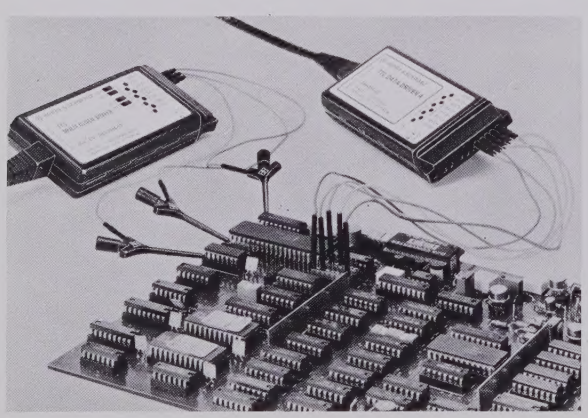
Control of microprocessorized subassemblies, bussed systems and logic circuits

Data sheet
344 001
E-3

SPECIAL FEATURES

- 32 data channels with 1024 words
- Bit-serial data output with variable word length and memory depth up to 16,384 bits
- Internal and external clock up to 10 MHz
- Tristate words programmable by channel groups
- Multi-clock output for 4 strobe signals
- Ease of operation due to module-oriented data management with
  symbolic names for fetching data sets,
  line-oriented editor for easy variation, deletion, duplication, etc. of data
- Data format either binary, octal, decimal or hexadecimal
- Possibility of linking data sets to form module sequences
- Data output with programmable pauses
- Separate drivers for low-noise test-item supply via short lines
- Problem-free test-item adaptation thanks to large variety of accessories
- Programming via IEC bus
- 3-wire handshake conforming to IEC 625-1 standard and 2-wire handshake with data rates up to 1 MHz
- Data storage on magnetic-tape cassettes with option

Logic Generator IGA with clock driver (including driver adapter, cable and miniclips; all of the plug-in type) and data driver



Test-item adaptation via miniclips: the driver adapter (including the cables) plugged onto the driver case permits rapid change of test items

General

In addition to the typical evaluation equipment, such as logic and signature analyzers, in use for many years, the analysis of logic signals requires **bit pattern generators** for the correct driving of digital circuits and systems to be tested. The functions of a bit pattern generator far exceed those of conventional word generators and cover

- stimulation of bidirectional data lines
- simulation of the different clock signals of microprocessors
- supplying a sufficient number of data channels
- generation of bit-serial data with high clock frequency

Moreover, it is necessary to check the digital control and setting of analog and other technical systems. In this case the form of the bit patterns produced is less important than their control function or meaning. **Information generators** have to meet additional requirements regarding their ease of operation:

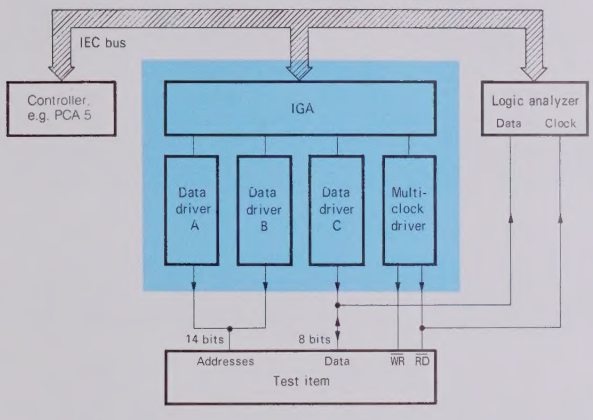
- simple data entry and modification
- easy fetching and outputting of user-oriented bit patterns (using plain text names)

Characteristics and uses

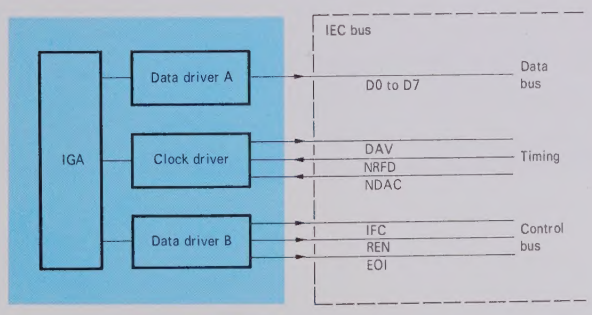
The **Logic Generator IGA** (see photos to the left) combines all the characteristics listed above and can thus be used both as a bit pattern generator and an information generator. It consists of the generator proper and of plug-on drivers.

Simplified generation of bit patterns Thanks to its great ease of operation, the IGA permits bit pattern generation with considerably less outlay. The editor functions and the cassette drive afford completely new applications especially in the development of digital circuits with reduced time and cost factors. The **IEC-bus interface** makes the IGA system-compatible.

Testing of bus-controlled functions The IGA is especially suitable for stimulating microprocessorized systems in its **multi-clock mode** and for functional testing of bussed systems by programming **tristate words** in channel groups (see diagrams below). In contrast to the conventional data and word generators, it thus covers all the applications involved in the stimulation of digital circuits.



Digital functional test in multi-clock mode (using IGA, logic analyzer and process controller)



Test of IEC-bus listener

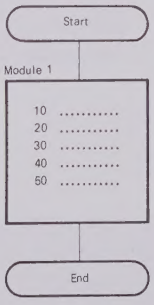
Used to replace rigged-up generators Thanks to the programmability of its functions, the IGA can be adapted to each measurement task without the need to resort to special, test-item-oriented solutions, thus saving time and reducing costs. Moreover, the use of a batch-produced instrument **reduces the risk of erroneous measurements** (and thus expensive down times). The risk of erroneous measurements due to unsatisfactory data retention is high in digital engineering. Here, in particular the variety of drivers available for the IGA (see pages 2 and 6) offers a high degree of safety.

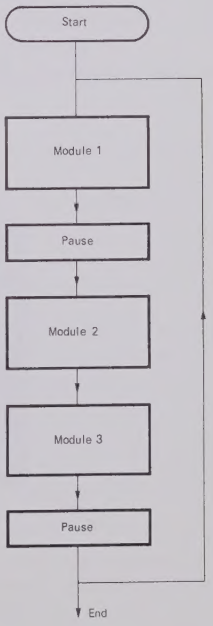
Technical details

Operation The sloped keyboard which is divided into edit, data and run sections permits convenient entry of the desired bit patterns and activation of the data output. In the display section, the module name or line number appears on the left and the data are read out on the right.

Data management The IGA uses **module-oriented** data management. A module consists of numbered data lines and is stored under a selectable, symbolic name; a maximum of 25 modules can be stored. By addressing the line numbers or calling up the module names, the data words or data blocks (modules) can be easily output (see diagrams below), varied, erased, duplicated, etc. By linking modules it is possible to obtain **module sequences** (with pauses) which are assigned a name of their own.

Data entry The data can be entered in the binary, octal, decimal or hexadecimal numerical system. The format instruction permits the channels of the IGA to be divided into a maximum of four groups, each being able to handle a different numerical system. The kill instruction deletes entire data words. A **cursor** enables individual figures to be changed. To extend the existing data set, additional lines can be inserted between lines previously entered.

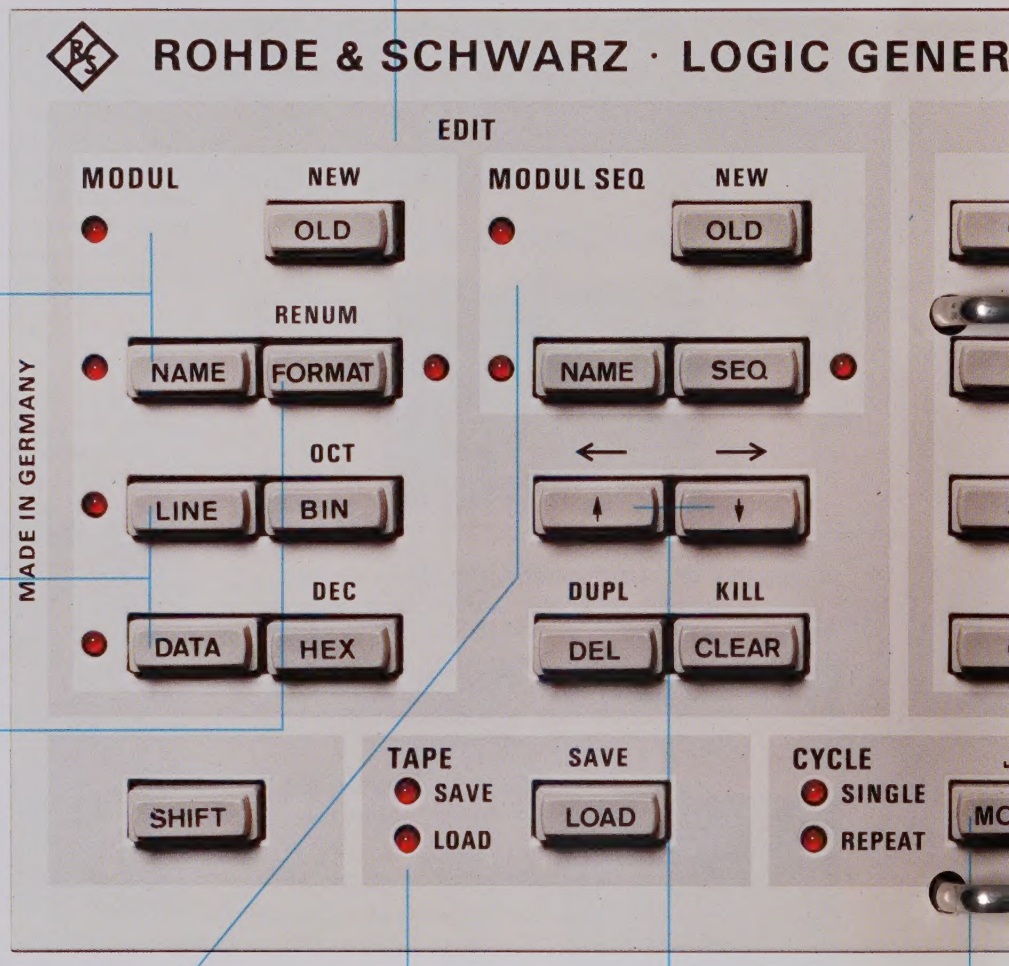




Left: data output of a single module (cycle mode: single)

Right: repetitive data output of a sequence with pauses (cycle mode: repeat)

LOGIC GENERATOR IGA



Convenient editing capability

Management of data sets under symbolic names

Easy entry, variation and expansion of data using line numbers

Maximum four different data formats per line: binary, octal, hexadecimal or decimal system

Linking of data sets to obtain sequences

Data storage and loading with cassette drive

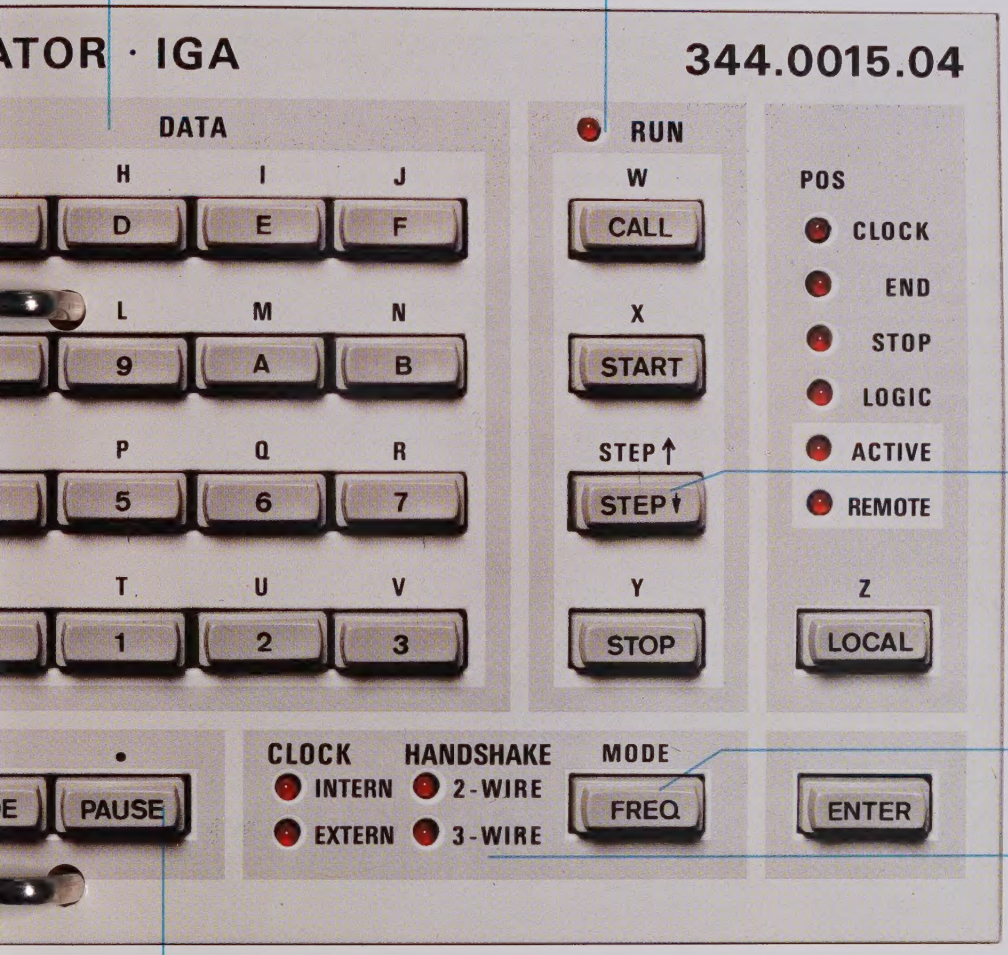
Cursor-aided data entry

Single and repetitive output of data sets

KEYBOARD and REAR PANEL

Entry of data
and module names

Easy checking
of data output



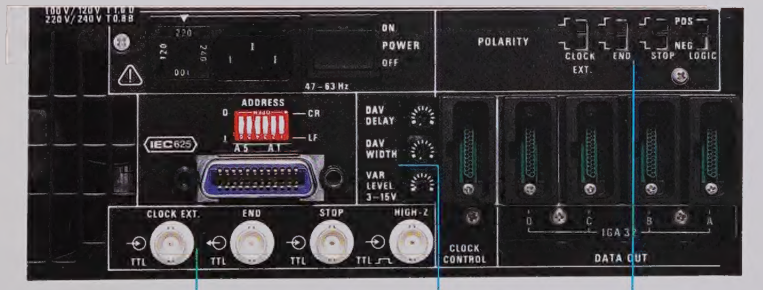
Scale
1:1

Single-step
data output

Programming of
internal clock frequency

Data output
with internal/
external clock,
in 2-wire or
3-wire handshake mode

Programmable
pause

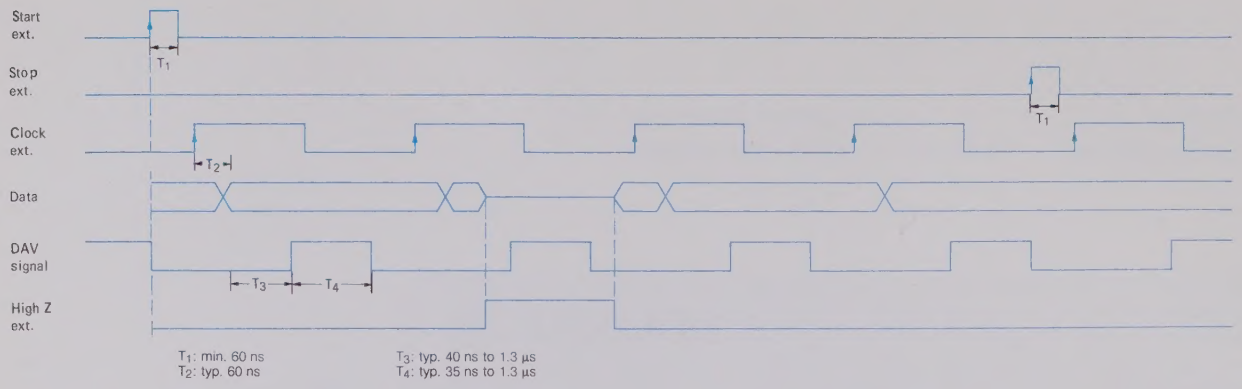


Control
connectors

Adjustment of DAV pulse
and high level

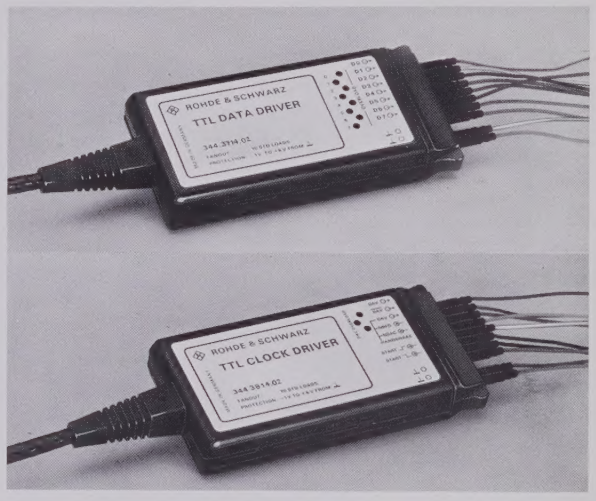
Polarity
switchover

CHARACTERISTICS



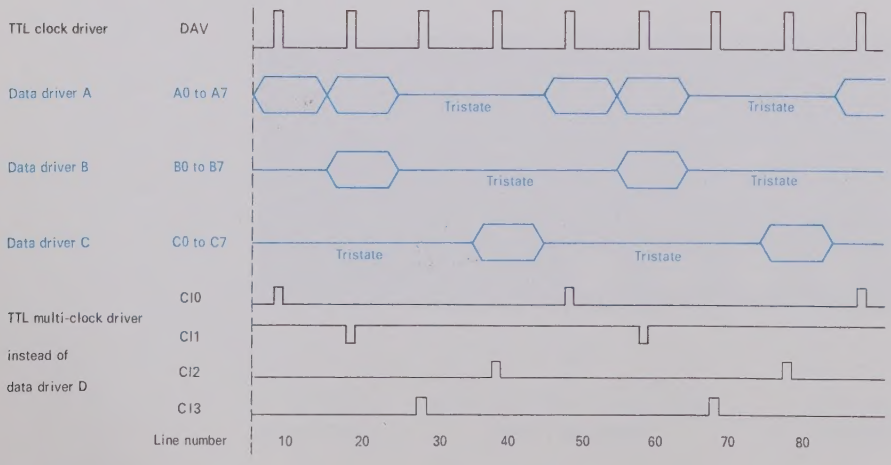
Timing diagram of data output (single module)

Data channels, adaptation The IGA has 32 data channels and a storage capacity of 1024 words, thus enabling long data sequences to be stored. Battery backup ensures data retention when the IGA is switched off. Separate data drivers permit the data outputs to be connected in groups of eight channels to the test item (see righthand column and page 2). In addition to general-purpose drivers with **adjustable levels** (IGA-Z1), **TTL** drivers (IGA-Z4) are available. The driver adapters with individual plug-on cables are connected to the driver case so that the test items can be easily changed. To permit low-noise signal application, these cables are only 30 cm long and can be plugged onto wire-wrap pins (0.8 mm). IC (dual-in-line) adapters are supplied with the IGA. IC test clips and miniclips are available as accessories. The data output can be clocked, single-stepped, single-shot and repetitive with programmable, variable pauses (see diagrams above and on page 3).



TTL Data Driver IGA-Z4 (top) and TTL Clock Driver IGA-Z5 (bottom)

Programming of tristate words (see diagram below). For driving bidirectional bussed systems, the data channels can be switched in groups of eight lines from the active to the high-impedance state during the output. At the end of the data output, the channels can also be switched in groups to high impedance. In the tristate mode, the IGA can, for instance, instruct a subassembly to transfer data and then have its data lines go high impedance whereas the address lines remain active. The DAV line continues to supply the clock for the talking subassembly.



Example for tristate mode (blue) and multi-clock mode covering several lines

Clock output The IGA can be used with a built-in or an external **clock generator** up to 10 MHz. In addition, 3-wire handshake (corresponding to source handshake in accordance with IEC 625-1) and 2-wire handshake up to 1 MHz (with TTL drivers) can be selected. It is possible to set the start and width of the clock pulse. The Clock Driver IGA-Z2 has **adjustable levels;** the Clock Driver IGA-Z5 delivers **TTL levels.** For the connection of the clock drivers see under data channels.

Generation of serial data To generate bit-serial data, a driver (IGA-Z9) is available serializing up to 16 data channels of the IGA. In this way bit-serial data with word lengths freely adjustable between 2 and 16 bits can be produced. The available memory depth permits up to 16,384 bits to be stored depending on the word length. Data are output with TTL levels. When using the external clock input, the data rate can be increased to up to 30 MHz for word lengths of more than 3 bits.

Multi-clock output The stimulation of microprocessorized systems is possible with the aid of the multi-clock driver (IGA-Z6, see below; signal examples on preceding page, bottom) delivering **TTL levels.** The driver provides data for four channels and, in addition to the continuous clock signal, four separately programmable strobe signals whose polarity can be selected individually on the driver case. The number of data channels available is thus reduced to 28.

In conjunction with the Logic Generator IGA, the **Analog Driver IGA-Z11** enables the generation of analog signals of virtually any waveform such as they cannot be produced

TTL Multi-clock Driver IGA-Z6

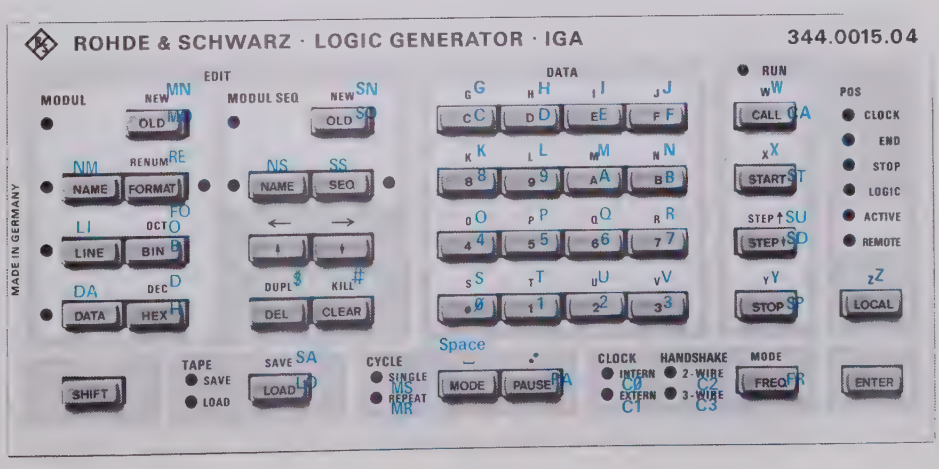
Cassette compartment of Cassette Drive Option IGA-B2

with common function generators. A 12-bit D/A converter in the Analog Driver converts the digital patterns stored in the IGA into analog signals.

The amplitude of the output signal can be continuously varied with a potentiometer. In addition, an offset adjustable between +5 V and −5 V can be selected.

Cassette drive (option; for illustration see above) Two times seven complete IGA memory contents, corresponding to the capacity of the cassette, or individual data sets can be stored on magnetic-tape cassettes (for instance for setting up a bit pattern library) and subsequently be read in again. In this way, the IGA can be used simultaneously for different tasks without losing data.

IEC-bus interface This interface makes the IGA programmable to IEC 625-1 with the front-panel keys (see photo below) and permits the logging of bit patterns on a printer. Moreover, data patterns generated in a desktop computer can be entered into the IGA via the IEC bus. In this way, the IGA acts as an intelligent, high-speed output interface for computers.



Association of programming commands (blue) with front-panel keys of IGA

SPECIFICATIONS

Data memory

Storage capacity	32 channels with 1024 words
Storage time (at + 25 °C)	approx. 6 months (battery-buffered)

Data entry

Format	binary, octal, decimal or hexadecimal; can be used simultaneously in one data line
Data expansion and variation	after fetching the freely selectable symbolic module name or a line number; editor-aided
Number of storable modules	max. 25
Display	data: 16 digits (7-segment indicators), module name and line number: 6 digits (LED matrix)

Data output

Modes	single-step/single-shot/repetitive with programmable, variable pauses
Start and stop	by keystroke or external TTL level

Data drivers (data signal outputs)

Number of data drivers connected	4

General-purpose Data Driver IGA-Z1

Channels	8
Levels	can be set on rear panel of IGA
Low	≦ 0.8 V
High	3 to 15 V
Fan-out (for high = 5 V)	1 standard TTL load
Capacitive loading	< 50 pF

TTL Data Driver IGA-Z4

Channels	8
Level	TTL
Fan-out	10 standard TTL loads
Permissible overloading	− 1 V/+ 6 V
Continuous overloading	simultaneous overload on 2 channels max.
Rise and fall times	typ. 30 ns (with 10 standard TTL loads)

Tristate words	programmable for switching data channels to high impedance in every line
Switchable channels	8-bit groups
Maximum data channels available when using tristate words	28
End tristate	switching the data channels to high impedance at the end of the output
Control	internal or external

Serial Data Driver IGA-Z9	can be used instead of data drivers A, B and C, D
Channels	1 (2 outputs)
Level	output 1: TTL output 2: current loop
Word length	2 to 16 bits, adjustable
Clock frequency	2 Hz to 10 MHz for internal clock, 0 to 30 MHz for external clock with word length of at least 3 bits
Control inputs	
External clock	TTL (negative logic)
Ready	TTL (positive logic)
Control outputs	
Bit synchronization	TTL (negative logic), bit association switchable
Connectors	BNC

Analog Driver IGA-Z11	can be connected instead of 2 data drivers for parallel data
Resolution	12 bits
Level setting	with LEVEL control
0-dB output	max. ± 10 V (EMF)
Max. load current	4 mA
Source impedance	50 Ω
-20-dB output	± 1 V (EMF)
Source impedance	50 Ω
Offset	can be disabled
Setting range (referred to 0-dB output)	− 5 V to + 5 V
Bandwidth	100 kHz

Clock control

Modes	internal/external/2-wire handshake/3-wire handshake (IEC 625-1)
Internal clock	1 Hz to 10 MHz
Setting resolution	2 digits
Frequency error	< 0.1 %
External clock	0 to 10 MHz
Active edge	can be set on rear panel of IGA
Handshake clock	0 to 1 MHz
Wait times	programmable
Interval between end of one cycle and beginning of next	0 to 60 s (resolution 1 ms)
Interval between data blocks	1 to 60 s, can be disabled (resolution 0.1 s)

Clock drivers (clock signal outputs)

General-purpose Clock Driver IGA-Z2

Synchronous clock outputs	DAV, inverted DAV
Levels	can be set on rear panel of IGA
Low	≦ 0.8 V
High	3 to 15 V
Fan-out (for high = 5 V)	1 standard TTL load
Capacitive loading	< 50 pF
Asynchronous control lines (for 2-wire and 3-wire handshakes)	DAV, NRFD, NDAC

TTL Clock Driver IGA-Z5

Synchronous clock outputs	DAV, inverted DAV
Level	TTL
Fan-out	10 standard TTL loads
Asynchronous control lines (for 2-wire and 3-wire handshakes)	DAV, NDAC, NRFD (resistor wiring as per IEC-625-1)
TTL Multi-clock Driver IGA-Z6	can be connected to B and D instead of a data driver
DAV signal outputs	4; separately programmable
Polarity	can be separately changed over
Level	TTL
Fan-out	10 standard TTL loads
Data outputs	4 (same as on TTL data driver)
Delay of DAV signal with respect to data	80 ns to 1 µs } can be set on
Width of DAV signal	40 ns to 1 µs } rear panel of IGA
Control inputs and outputs	TTL level

Inputs:

Start	start of data output	on clock driver
Start	start of data output (inverted signal)	on clock driver
Stop	stop of data output	on rear panel of IGA; BNC
Clock ext.	clock input	on rear panel of IGA; BNC
High Z	tristate	on rear panel of IGA; BNC

Output:

End	end of data output	on rear panel of IGA; BNC

Cassette Drive IGA-B2 (option)

Capacity of cassette	2 × 7 IGA memory contents

Remote control

System	IEC 625-1
Connector	24-way (Amphenol)
Interface functions	T5 talker L4 listener RL1 remote/local DC1 device clear DT1 device trigger SR1 service request

General data

Rated temperature range	+ 5 to + 45 °C
Storage temperature range	− 20 to + 70 °C
AC supply	100/120/220/240 V ± 10 %, 47 to 63 Hz (70 VA)
EMC	to DIN 57 875 (VDE 0875): RF suppression N − 12 dB
Dimensions, weight	325 mm × 110 mm × 420 mm, 5 kg

Ordering information

Order designation	▶ Logic Generator IGA 344.0015.04

Accessories supplied
Dual-in-line adapter, 14-, 16-, 20-, 24-, 40-contact
Carrying case, power cord, manual

Options

Cassette Drive (including 6 cassettes 344.3114.02)	IGA-B2	344.2918.02

Recommended extras

General-purpose Data Driver	IGA-Z1	344.3414.02
General-purpose Clock Driver	IGA-Z2	344.3514.02
Adapter	IGA-Z3	344.3614.02
TTL Data Driver	IGA-Z4	344.3714.02
TTL Clock Driver	IAG-Z5	344.3814.02
TTL Multi-clock Driver	IGA-Z6	344.3914.02
Analog Driver	IGA-Z11	356.8700.02
Mini-cassettes	IGA-Z7	344.3114.02
Set of Miniclips (20)[1]	IGA-Z8	344.1905.02
Serial Data Driver	IGA-Z9	344.3143.02
IEC test clips, dual-in-line, 14-contact		099.5222.00
16-contact		099.5239.00
20-contact		099.5268.00
24-contact		099.5245.00
40-contact		099.5251.00
Process Controller	PCA 5	375.2010.02

[1]) Supplied with the drivers

ROHDE&SCHWARZ GmbH & Co. KG. · D-8000 München 80 · Mühldorfstr. 15 · Tel. (089) 41 29-10 Int. + 49 89 41 29-10 · Telex 5 23 703
Printed in the Fed. Rep. of Germany · Subject to change · Data without tolerances: order of magnitude only
1085 (Bv)

1 Characteristics and Uses

The Logic Generator IGA is used for stimulation of circuits provided with digital interface such as microprocessor peripherals or digitally driven analog circuitry.

32 data channels are available. The storage capacity is 1024 words. The data memory is battery-buffered so that its contents are not lost when the instrument is switched off.

The data outputs, each of 8 bits, are combined in data drivers and accommodated in cabinets of their own. Independently of the remaining data drivers, each data driver can be used in the tristate mode, thus permitting data output on bidirectional bussed systems.

The clock driver, which is accommodated in a separate cabinet like the data drivers, includes the driver circuitry for the DAV signal. This signal confirms that the data are valid and can be used as a strobe or clock pulse for the stimulated circuitry. Compared with the data, the DAV signal is slightly delayed. The delay and the width of the pulse can be set on the rear panel of the IGA.

If one DAV signal is not sufficient, a multi-clock driver can be used instead of a data driver delivering programmable DAV signals on 4 channels instead of data.

The clock for the data output is internally produced by a synthesizer which can be set to frequencies between 1 Hz and 10 MHz. Moreover, data output is also possible with an external clock or in the handshake mode.

The IGA can be used with single-shot (SINGLE) or repetitive (REPEAT) data output. It is possible to define pauses during which the IGA stops the data output after each cycle. The data entry and data management of the IGA are similar to those of desktop computers. Like a BASIC statement, each data word receives a line number. The line number and the data word constitute together a data line. Several data lines constitute a module. Thanks to the line numbers, the IGA editor can address each data line and it is easy to erase, vary or duplicate data lines or to insert new lines.

Each module is defined by a name under which it is stored in the IGA like a file in a computer. Thus several data sets (modules) can exist in the IGA without affecting each other. Before entering the data, the user must enter the module name.

An exception is the direct module which does not need to be given a name since it is automatically provided when the IGA is switched on. Simultaneously, DIRECT appears on the display.

All the data entered in this case are saved under the name DIRECT and are available as long as the IGA remains switched on. When the set is switched off, all the data entered into the direct module are lost unless they have previously been stored under a different name (section 2.2.2.5) with the aid of the DUPL key 22 (Fig. 2-2). Only then are the data retained in the battery-buffered CMOS RAM of the IGA so that they can be called up unter the new name later. Using the direct module is recommended in particular for the novice, so that he can get acquainted with the instrument; it is also suitable for data sets which are not to be stored in the IGA. This avoids having later to erase old data sets whose presence reduces the storage capacity of the IGA.

If more than one module is to be stored simultaneously in the IGA, or if the data are not to be lost in the case of a power failure, modules defined by names are used. For module name entry see section 2.2.2.

In certain cases, the IGA is required to output a number of modules, one after another. For this purpose, a list, i.e. the sequence, is entered into the IGA. This list contains the names of the individual modules in the order in which they are to be output again. Like the individual modules, the module sequence is defined by a name, can be called up by its name and can be applied to the data drivers with the aid of the START key 8. For further details on the module sequences see section 2.2.3.

2 Preparation for Use and Operating Instructions

2.1 Preparation for Use

2.1.1 Legends

(see Figs. 2-2 to 2-4)

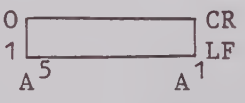
No.	Inscriptions	Function
1	MODULE OLD	By pressing this key, entering the module name and finally pressing the ENTER key, a stored module can be fetched.
	NEW	By pressing this key, entering the module name and finally pressing the ENTER key, a new module can be created.
2	MODULE SEQ NAME	Same effect as MODULE NAME key (see 31), only for module sequences.
3	MODULE SEQ OLD	By pressing this key, entering the module sequence name and finally pressing the ENTER key, a stored sequence can be fetched.
	NEW	By pressing this key, entering the new module sequence name and finally pressing the ENTER key, a new module sequence can be created.
4	SEQ	Causes display of the name of the first module in the sequence, replacement of the first module by entering the name of another module (the name entry being terminated by pressing the ENTER key), and display of the name of the second module in the sequence, etc.
5	DATA	Keyboard for entry of data and names.
6	CALL	By pressing this key and entering the function name, a special function can be called. Examples: outputting the data on an IEC-bus-compatible printer; or formatting the magnetic tape.
	W	If the SHIFT key is pressed, this key allows to enter the name.

No.	Inscriptions	Function
7	RUN	The LED indicates that the IGA is reading out data (lights up when the START key is pressed and goes out as soon as the STOP key is pressed).
8	START X	Starts data output. If the SHIFT key is pressed, this key allows to enter the name.
9	CLOCK	The LED indicates that the POL RITY CLOCK EXT. has been switched to the positive-going edge.
10	END	The LED indicates that POLARITY END has been switched to the positive-going edge.
11	STOP	The LED indicates that POLARITY STOP has been switched to the positive-going edge.
12	LOGIC	The LED indicates that POLARITY LOGIC has been switched to POS.
13	ACTIVE REMOTE	The LED is on during data output and off during pauses. The LED indicates that the unit is in the remote-control mode.
14	LOCAL Z	As long as no local-lockout instruction has been sent, the REMOTE mode can be interrupted during the program run by pressing the LOCAL key. The next time the IEC-bus controller sends an address, the REMOTE mode is restored. If the SHIFT key is pressed, this key allows to enter the name.
15	ENTER	Causes data to be transferred, e.g. at the end of a data entry (all the keys with figures and letters as well as the keys HEX, BIN, DEC, OCT, "." and "␣").
16	STEP ↑ STEP ↓	Switches to the preceding or subsequent data line respectively, in the single-step data output control mode.

No.	Inscriptions	Function
17	STOP	Stops data output.
	Y	If the SHIFT key is pressed, this key allows to enter the name.
18	CLOCK HANDSHAKE FREQ	Causes the frequency set to be displayed; the frequency can be varied by overwriting with new values.
	MODE o INTERN o 2-WIRE o EXTERN o 3-WIRE	Selects internal or external clock, 2-wire or 3-wire handshake; the LEDs indicate the mode selected.
19	CYCLE PAUSE	Causes the pause set for repetitive data output to be displayed; the pause can be varied by overwriting with new values.
	.	If the SHIFT key is pressed, this key allows to enter data.
20	CYCLE MODE	Switches to repetitive or single-shot data output; the corresponding LED indicates the mode selected.
		If the SHIFT key is pressed, this key can be used for entering data.
21	→	Cursor key Used for changing names and data that are being displayed.
	KILL	Key for erasing data lines. The data line which is erased is the one which is being indicated. Requires confirmation by the ENTER key (for erasing modules and module sequences, see NAME key).

No.	Inscriptions	Function
22	←	Cursor key Allows to change names and data that are being displayed.
	DUPL	Writes into the data line whose line number is being indicated the datum (item of data) from the line whose line number has been entered after pressing the DUPL key. Requires confirmation by the ENTER key (for the duplication of modules and module sequences, see NAME keys 2 and 31).
23	LOAD SAVE	Loads and saves data on magnetic tape.
24	SHIFT	If the SHIFT key is pressed, the keys have the function marked above each key on the front panel.
25	HEX DEC	Numbering system for data entry, if the FORMAT key has been pressed.
26	DATA	Causes display of the first data line of a module; the data line can be changed by overwriting the display with new data and confirming the entry by the ENTER key. Simultaneously the next data line is displayed, and so on.
27	BIN OCT	Numbering system for data entry; first the FORMAT key has to be pressed.
28	LINE	Causes display of a data line (requiring the entry of a line number and confirmation of the entry by the ENTER key). New data lines can be inserted, or old data lines modified, by overwriting with new data.
29	↑	Cursor key Switches to the preceding data or name line.
	DEL	Deletes the character displayed at the cursor position; moves the cursor one digit to the left.

No.	Inscriptions	Function
29a	↓	Cursor key Switches to the subsequent data or name line.
	CLEAR	Key for erasing data lines. The data line being displayed is erased. Requires confirmation by the ENTER key (for erasing modules or module sequences see NAME key).
30	FORMAT	Causes display of the format selected. The format can be changed by overwriting the display with new values.
	RENUM	Causes renumbering, in increments of ten, of the module selected, after the ENTER key has been pressed.
31	MODULE NAME	Causes display of the module name. The module name can be changed by overwriting the display with a new name and confirming the entry by means of the ENTER key. The menu of all the modules stored in the IGA can be obtained with the ↓ ↑ keys. The module whose name is displayed - is simultaneously made the current one, - can be erased by pressing the KILL key and subsequently the ENTER key, - can be duplicated by pressing the DUPL key, entering a new name and finally pressing the ENTER key.
32		Display section
33		Cassette drive (option) (see Fig. 2-4)
34	47-63 Hz	AC supply connector, voltage selector
35	ON POWER OFF	Power switch

No.	Inscriptions	Function
36	DAV DELAY DAV WIDTH VAR LEVEL 3-15 V	Potentiometers for: - DAV delay - DAV pulse width - voltage setting for general-purpose data drivers
37	POLARITY CLOCK EXT. END STOP LOGIC	Switches for polarity reversal: - switch for external clock signal - switch for end signal - switch for stop signal - positive or negative logic
38	DATA OUT	Connectors for data drivers
39	CLOCK CONTROL	Connector for clock driver
40	HIGH-Z TTL	Input for tristate control signal
41	STOP TTL	Input for stop signal
42	END TTL	Output for end signal
43	CLOCK EXT. TTL	Input for external clock signal
44	IEC 625 ADDRESS 	24-way IEC-bus connector Switch for setting the device address and the terminating character
45		Blower

2.1.2 Adjusting the Set to the Available AC Supply Voltage

The IGA can be operated from an AC supply of 100 V, 120 V, 220 V or 240 V. It is factory-set for 220 V. To adapt it for operation from a different AC supply voltage, withdraw the voltage selector, insert the fuse required for the AC supply voltage available and replace the voltage selector such that the arrow (upper side of the opening) points to the marking of the new voltage value.

The following fuses are required:

220 V and 240 V: T 0,8
100 V and 120 V: T 1,6

2.1.3 Connecting the Data Drivers and the Clock Driver

The data drivers and the clock driver are connected to the rear panel. The CLOCK CONTROL connector 39 (Fig. 2-3) is used for the clock driver and the DATA OUT connectors 38 for the data drivers. Potentiometer 36 can be used to set the output high voltage of the general-purpose data drivers.

If the multi-clock driver is used instead of a data driver, it must be connected to DATA OUT D 38. When connecting the drivers, the IGA must be switched off.

2.1.4 Switching the Set on

The IGA is connected to the AC supply at 34 via the power cable provided. The IGA is switched on by means of the power switch located on the rear panel. Then the Logic Generator carries out a self-test, briefly indicates the IEC-bus address and then signals DIRECT on the display.

2.2 Manual Operation

2.2.1 Direct Module

A certain number of data lines entered under a name by the user and handled like files in a computer is called a module.

However, the direct module does not need any name entry since it is automatically provided when the IGA is switched on. Simultaneously, DIRECT appears on the display.

In contrast to all the other modules stored in the IGA, the contents of the direct module are lost when the IGA is switched off.

2.2.1.1 Introduction

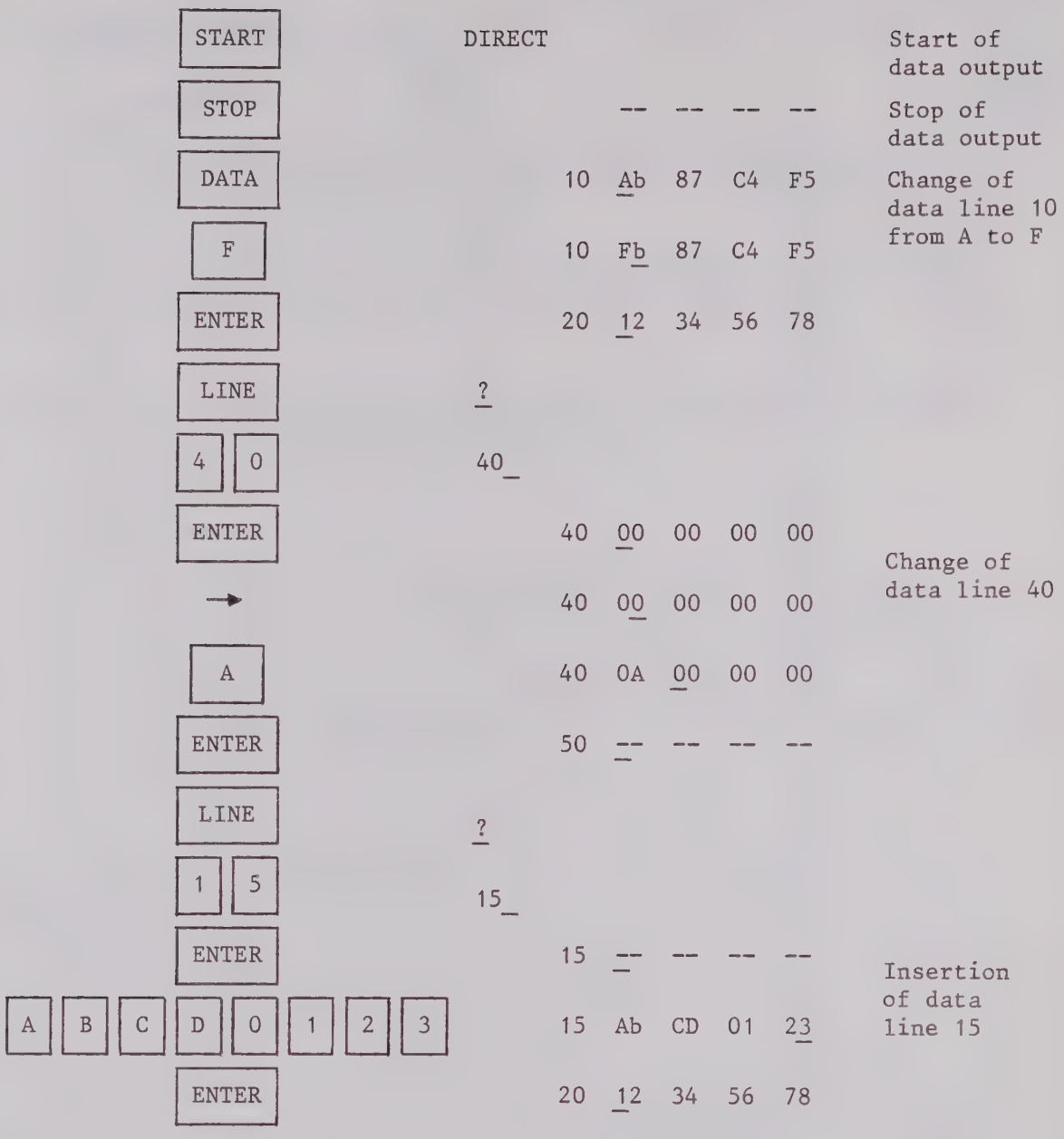
When the IGA is switched on, it automatically selects certain operational settings. The output frequency is set to 10 kHz and the end tristate mode is disabled. Also, hex format is specified for the data entry, i.e. the IGA expects two hex numbers to be entered for each data driver. The following sections describe how these standard settings can be changed. However, the following example is based exclusively on these standard settings.

When the DATA key is pressed, the number 10 - the number of the first data line - appears on the left of the display. Now data can be entered. The LSB appears on the right-hand side of the display corresponding to pin 0 of data driver A. Entry errors can be corrected with the keys ⟵ , ⟶ and DEL as well as by overwriting with new data. If the display shows the desired data, they can be confirmed with the aid of the ENTER key 15. This causes the data to be stored away and the IGA indicates the next line number, in this case 20. Now the next data line can be entered, etc. The application of the data to the data drivers is caused by the START key 8.

The output can be stopped with the STOP key 17. If additional data are to be entered, the DATA key 26 can be used to return to the first data line defined by the number 10. As during the first data entry, the data can be changed using the keys ⟵ , ⟶ and DEL. When the ENTER key 15 is pressed, the old data in the RAM are replaced by the new data on the display. Data lines with different line numbers can be reached either by way of the keys ↑ , ↓ or with the LINE key 28 requiring the entry of the desired line number and confirmation by the ENTER key. In this way, intermediate lines can be inserted, for instance line 15.

Example of Entry into the Direct Module

Entry	Display on the IGA					Remarks
POWER ON	IECADR DIRECT		12			
DATA	10	--	--	--	--	
A	10	A-	--	--	--	
B	10	Ab	--	--	--	
8	10	Ab	8-	--	--	
7	10	Ab	87	--	--	
C	10	Ab	87	C-	--	Entry of a data line
3	10	Ab	87	C3	--	
F	10	Ab	87	C3	F-	
1	10	Ab	87	C3	F1	
5	10	Ab	87	C3	F5	
←	10	Ab	87	C3	F5	
←	10	Ab	87	C3	F5	Correction of a data line: changed from 3 to 4
4	10	Ab	87	C4	F5	
ENTER	20	--	--	--	--	
1 2 3 4 5 6 7 8	20	12	34	56	78	
ENTER	30	--	--	--	--	Entry of further data lines
A B C D E F 0 1	30	Ab	CD	EF	01	
ENTER	40	--	--	--	--	
0 0 0 0 0 0 0 0	40	00	00	00	00	
ENTER	50	--	--	--	--	


START
DIRECT
Start of data output
STOP
-- -- -- --
Stop of data output
DATA
10 Ab 87 C4 F5
Change of data line 10 from A to F
F
10 Fb 87 C4 F5
ENTER
20 12 34 56 78
LINE
?
4
0
40_
ENTER
40 00 00 00 00
Change of data line 40
→
40 00 00 00 00
A
40 0A 00 00 00
ENTER
50 -- -- -- --
LINE
?
1
5
15_
ENTER
15 -- -- -- --
Insertion of data line 15
A
B
C
D
0
1
2
3
15 Ab CD 01 23
ENTER
20 12 34 56 78

2.2.1.2 Entry

FORMAT

2.2.1.2.1 Entry of Data Format

By entering the format, the user defines the numbering system which he is going to employ for data entry. The binary, octal, decimal or hexadecimal system can be used. The 32 channels of the IGA can be divided into up to four groups, each of which can be used with a different data format. A number, of one or 2 digits, is entered for each of these groups determining the number of channels in that group; then one of the keys BIN 27, OCT 27, HEX 25 or DEC 25 is pressed. The numbers for hexadecimal entry must be multiples of 4, those for the octal entry multiples of 3 and those for the decimal entry greater than 3. The format entry, which is on the right-hand side of the display, refers to the LSB. If the sum of all the format specifications is smaller than 32 bits, the remaining channels at the MSBs are filled up with zeros. The standard setting is 8H 8H 8H 8H. When the FORMAT key 30 is pressed, the IGA displays the data format selected. The format can be changed with the keys ⟵ , ⟶ , DEL and by overwriting with new characters. However, the old format specified is replaced by the new format being displayed only after the ENTER key 15 has been pressed.

Examples:

Key pressed	Display on the IGA
FORMAT	FORMAT _8H 8H 8H 8H (standard setting)

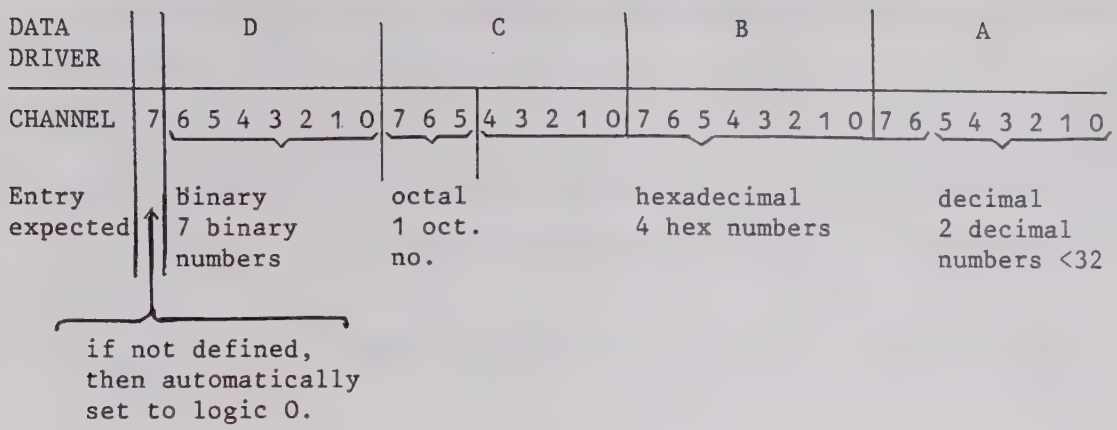
Data format produced:

DATA DRIVER	D		C		B		A		
CHANNEL	7654	3210	7654	3210	7654	3210	7654	3210	LSB
Entry expected	1 hex number	1 hex number	1 hex number	1 hex number	1 hex number	1 hex number	1 hex number	1 hex number	

Example of Format Entry

Key pressed		Display on the IGA					Remarks
FORMAT		FORMAT	_8H	8H	8H	8H	Standard setting
→		FORMAT	8H	8H	8H	8H	
7		FORMAT	7H	8H	8H	8H	
BIN		FORMAT	7b	_8H	8H	8H	
→		FORMAT	7b	8H	8H	8H	
3		FORMAT	7b	3H	8H	8H	
SHIFT	OCT	FORMAT	7b	30	_8H	8H	
1		FORMAT	7b	30	18H	8H	
6		FORMAT	7b	30	16H	8H	
→		FORMAT	7b	30	16H	_8H	
→		FORMAT	7b	30	16H	8H	
5		FORMAT	7b	30	16H	5H	
SHIFT	DEC	FORMAT	7b	30	16H	5d	
ENTER		HIGH - Z -					See 2.2.1.2.10 tristate control. Select only if end tristate is desired.

Data format produced for 7 BIN, 3 OCT, 16 HEX, 5 DEC



Note: Even for data outputs not used, a format has to be entered.
When entering data, it must be borne in mind that entry positions are also offered for data outputs not used.

Example: Data driver B not connected, with 05H to be output on data driver A and 07H on data driver C.

Format: 8H 8H 8H

Data entry: 05 00 07

- 07 — data for data driver A
- 00 — data for data driver B (no meaning as driver not connected)
- 05 — data for data driver C

2.2.1.2.2 Entry of Data Lines

DATA

The data entry mode is selected by means of the DATA key 26. The number of the first data line appears on the left-hand side of the display. On the right-hand side, the contents of the data line or, if the line does not yet contain any data, information on the entered format is displayed. If a data line comprises more characters than the display is able to represent, the line is divided into a first and second line. The second line contains the LSBs.

Example: Format display: 7 BIN, 3 OCT, 16 HEX, 4 DEC

```
                                  BIN
10 1                              -

10 2  - - - - - - -   -  - - - -  - -
        BIN           OCT  HEX    DEC
```

As long as the cursor bar is blinking, data can be entered or changed by moving the cursor under the character which needs to be changed with the aid of the ←, → keys and by overwriting this character with the correct one. The DEL key 29 permits any incorrect characters to be erased. When a data line has been entered completely, it must be confirmed with the ENTER key 15. In this way, the contents of the display are stored and the next data line appears, with its line number, in the display. The ↑ , ↓ keys also permit data lines already confirmed by the ENTER key to be displayed again and modified.

Example: Entry of data lines (FORMAT 8H)

Key pressed		Display on the IGA	Remarks
DATA	10	--	
A	10	A	Entry of data line 10
B	10	AB	
ENTER	20	--	
0	20	0	
1	20	01	Entry of data line 20
ENTER	30	--	
4	30	4	
5	30	45	Entry of data line 30
ENTER	40	--	
↑	30	45	Display of data line 30
↑	20	01	Display of data line 20
8	20	81	Change of data line 20 and
ENTER	30	45	display of the next data line

2.2.1.2.3 Insertion of Additional Data Lines

LINE

If a line is to be inserted between two lines already entered, use the LINE key 28.

After pressing the LINE key, enter a line number located between the numbers of the old data lines and press the ENTER key 15. Then enter the new data line and confirm it with the ENTER key.

If, with the aid of the LINE key, a new data line is to be added at the end of the current data set, the new line number entered must not be greater than the next multiple of 10 following the last old line number.

Example: Data line 15 is to be inserted between data lines 10 and 20.

Key pressed		Display on the IGA	Remarks
LINE	?		
1	1 _		
5	1 5 _		
ENTER	1 5	_ - - -	Data line 15 does not yet contain any data
1	1 5	1 _ - -	
2	1 5	1 2 _ -	
3	1 5	1 2 3 _	
4	1 5	1 2 3 4	
ENTER	2 0	A b C d	Data line 15 is confirmed and the next line displayed

2.2.1.2.4 Correction of Data Entered Wrongly or by Mistake

Data can be corrected only when they are being displayed and when the cursor is blinking at the same time. For correction, the cursor is brought under the character which needs to be changed with the aid of the ←, → keys, and this character is then overwritten with the correct one. The DEL key 29 permits individual characters to be erased at the cursor position. The corrected data line is then confirmed by pressing the ENTER key 15, thus replacing the old, incorrect line by the new one.

If a data line which is not being displayed is to be changed, this data line can be made to appear on the display in the following way:

Key pressed	Display on the IGA	Remarks
LINE	?	
2	2 _	
8	2 8 _	Entry of the desired line number (e.g. 28)
ENTER	2 8 A b 7 8	Data line contents displayed

Another possibility of making lines to be changed appear on the display is to use the keys ↓ 29a and ↑ (29).

2.2.1.2.5 Cancelling Data Lines

KILL

Data lines can be cancelled only when they are being displayed. If the data line to be erased is not being displayed, it can be caused to appear on the display by pressing for instance the LINE key 28, entering the line number and pressing the ENTER key 15. Cancelling is initiated by the KILL key 21 (SHIFT →). The display then signals KILL??. If the KILL key had been pressed by mistake, this can be annulled with the CLEAR key 29a (SHIFT ↓). Otherwise, the data line can then be erased by pressing the ENTER key again.

Example: Cancelling data line 20

Key pressed	Display on the IGA		Remarks
LINE	?		
2	2_		
0	20_		Data line 20 is displayed
ENTER	20	8bEF9037	
KILL			
SHIFT →	KILL??		
ENTER	30	543210Ab	Data line 20 is erased and the next data line displayed

2.2.1.2.6 Duplication of Data Lines

DUPL

In certain cases, it may be desired that the IGA should output the same data line several times consecutively. Thanks to the DUPL key 22 (SHIFT ←), the tedious procedure of entering the same data line several times can be avoided. If the DUPL key is pressed and a line number - confirmed with the ENTER key 15 - is entered, the IGA produces in the current line the same data as included in the line with the number entered.

Example: The same datum as in data line 10 is to be entered in data line 16.

Key pressed		Display on the IGA		Remarks
	LINE	?		
	1	1_		
	6	16_		Data line 16 does not yet contain any data
	ENTER	16	--------	
	DUPL			
SHIFT	←	?	16	
	1	1_	16	
	0	10_	16	
	ENTER	16	846530AB	Brief display of the new data line 16 with the contents of data line 10
		20	4326ACE0	Display of the next data line (line 20 in the above example)

2.2.1.2.7 Renumbering of Data Lines

RENUM

The subsequent insertion of intermediate lines with the LINE key 28 produces line numbers which are not a multiple of 10. These numbers require additional storage locations in the IGA which reduces its effective storage capacity. The same holds true for erased lines whose line numbers are multiples of 10. Therefore it is recommended that from time to time the data lines be consecutively renumbered in increments of 10. Pressing the RENUM key 31 (SHIFT FORMAT) and the ENTER key 15 provides the data set with consecutive line numbers in increments of 10.

Key pressed	RENUM	Display on the IGA	Remarks
SHIFT	FORMAT	RENUM?	
	ENTER	READY	

2.2.1.2.8 Entering the Frequency

FREQ

The FREQ key 18 initiates the entry of the frequency with which the IGA is to output the data. Then the IGA displays the frequency currently set. The value displayed can be changed with the aid of the keys ← , → and DEL as well as by overwriting with new data. The numerical value can be entered with and without exponent, only the two highest-order digits being significant. The unit of measurement is Hz. Only after the ENTER key 15 has been pressed, is the old frequency value replaced by the new frequency displayed.

Example:

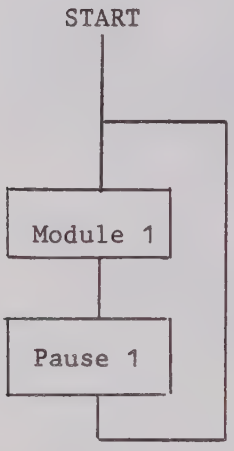
Key pressed	Display on the IGA		Remarks
FREQ	HERTZ	1.0E3	Display of the frequency currently set: 1000 Hz
2	HERTZ	2.0E3	
ENTER	HERTZ	2.0E3	New frequency: 2000 Hz

2.2.1.2.9 Entry of Pause

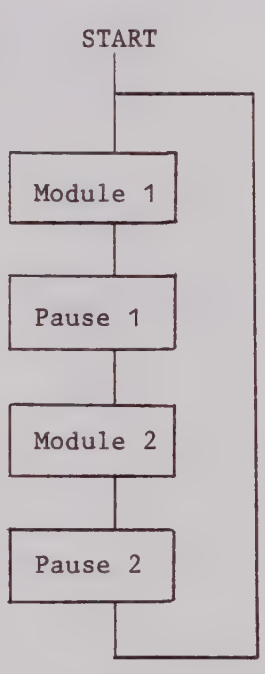
PAUSE

If a data set is being output repeatedly, the IGA can insert a pause at the end of the module and the module sequence before the module and the module sequence are output again. The entry of the pause length is initiated by the PAUSE key 19. The value is entered in the same way as the frequency value, but with four significant digits. The unit of measurement, and the shortest pause which can be set, is one millisecond. The entry requires confirmation by the ENTER key 15.

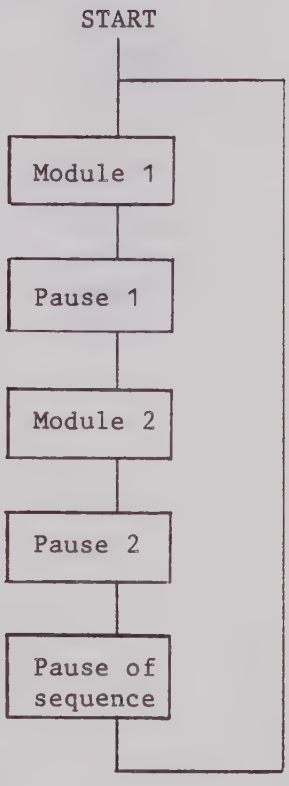
Pauses smaller than 400 ms are not taken into consideration for module sequences (see 2.2.3.2).



Individual module with pause



Module sequence with pauses of modules



Module sequence with module and sequence pauses

2.2.1.2.10 Tristate Mode

The IGA enables individual data drivers or a combination of several data drivers to be set to high impedance. This is essential for data output in bussed systems. Either line or end tristate mode can be selected.

In the end tristate mode, the data driver goes to the tristate mode after outputting the last data line.

In the line tristate mode, control bits permit the individual data drivers to be switched between the tristate and the normal mode in each data line. For this purpose, the most significant byte of the data memory (D7 to D4 of data driver D) is used so that only 28 data channels are available with line tristate.

The end tristate mode is selected together with the format entry. After the format entry has been confirmed with the ENTER key 15, HIGH-Z appears on the left-hand side of the display. Next the letters identifying the data drivers to be set to high impedance are entered. If the letter A is entered, data driver A is switched to tristate at the end of the data output. When AB is entered, this applies to the two data drivers A and B, and so on. This entry requires confirmation with the ENTER key. If no letters are entered, then none of the data drivers are set to high impedance.

The IGA can also be switched to tristate during the data output via the control input HIGH-Z 40 (on the rear panel). However, this signal affects only those data drivers for which end tristate has been entered.

Note: The line tristate mode is only operative during data output. If in the last line of a module a data driver is in the line tristate mode, the data outputs of the driver do not remain tristate at the end of the data output and the data of the preceding line are present at the outputs.

If a data driver is to remain in the tristate mode after data output, end tristate has to be entered for the driver in question.

The multi-clock driver cannot be set to the tristate mode.

Example: Data drivers A and C are to go to high impedance at the end of the data output.

Key pressed	Display on the IGA	Remarks
FORMAT	8H 8H 8H 8H	Display of the current format, in this case, the basic setting
ENTER	HIGH-Z _	
A	HIGH-Z A_	Entry of end tristate
C	HIGH-Z AC_	
ENTER	HIGH-Z AC	

The line tristate mode is selected together with the data entry. If a decimal point is entered instead of a datum, the corresponding data driver goes high impedance during data output.

FORMAT	DATA ENTRY	
4H 8H 8H 8H	b .. E7 03	Data driver C: tristate
4H 8H 8H 8H	b 03	Data drivers C and B: tristate
8H 30 5b	Ab 7 01010	No data driver in tristate mode
8H 30 5b	Ab	Data driver A: tristate
8H 60 8H 2b	AE 64 EF 01	No data driver in tristate mode

The decimal points entered for line tristate must be within the data drivers.

If, for instance, one data driver contains two bits of a hexadecimal number and the adjacent data driver the other two bits and if a decimal point is entered instead of a hexadecimal number, the data outputs are in the tristate mode only on the condition that all other entries relating to the two data drivers in question are decimal points.

Otherwise, the error message ERR 94 will be output.

In the same way as the data drivers, the DAV outputs on the clock driver can be caused to go high impedance in the line tristate mode. In this case, an L has to be entered instead of the datum. Line tristate for DAV is indicated

by a short bar in the upper right-hand section of the display 32 to the right of the line number. Simultaneously data channel D3 of data driver D is lost.

DAV tristate is cancelled by pressing the L key again.

2.2.1.3 Data Output

2.2.1.3.1 Data Drivers and Clock Driver

Various drivers are available which can be connected to the rear panel of the IGA. In addition to the TTL version fitted as a standard, there is a general-purpose version whose high-state output voltage can be adjusted with potentiometer 36. This version is suitable in particular for driving CMOS ICs.

Both the data drivers and the clock driver are provided in the two versions. The data drivers are used for outputting the data whereas the clock driver delivers a DAV signal and receives the NRFD and NDAC signals in the handshake mode.

2.2.1.3.2 Multi-clock Driver

In addition, a TTL multi-clock driver is available. Instead of a data driver, this driver can be connected to DATA OUT D 38. This multi-clock driver delivers DAV pulses, instead of data, on four channels; the polarity of these pulses can be set with a slide switch on the driver. The DAV signals are entered in the same way as the data. The format must be selected such that data can be entered in binary into channels D0, D1, D2, D3. A1 means that a DAV signal appears at the corresponding output of the multiclock driver. When a 0 is entered, no DAV signal appears. This applies for positive logic. If the switch on the rear panel is set for negative logic, a DAV signal appears when a 0 is entered.

Example: Positive logic

Key pressed	Display on the IGA	Remarks
FORMAT	4b 8H 8H 8H	
DATA	0001 AB E3 ..	Data are applied to the data drivers B and C. Data driver A is in the tristate mode
		A DAV pulse appears at the multi-clock driver output
DATA	0010 E3 06 45	Data are applied to data drivers A, B and C
		A DAV pulse appears at the multi-clock driver output CL1

2.2.1.3.3 Start, Stop of Data Output

START STOP

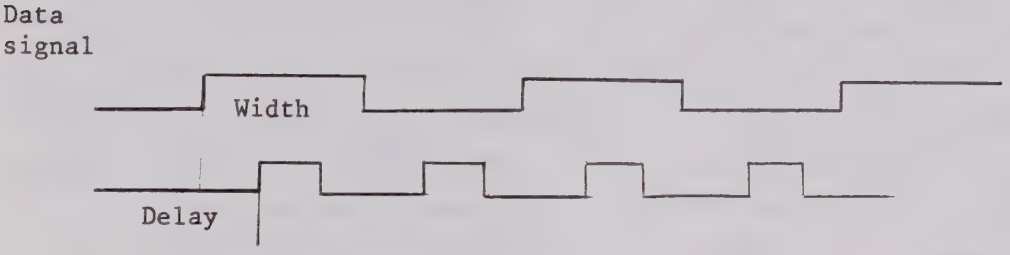
Either the START key 8 or a signal applied to the start input of the clock driver can be used to start data output. Either the STOP key 17 or a signal applied to the STOP input 41 on the rear panel can be used to stop data output. Then all data driver outputs are in the tristate mode. At the end of data output, the IGA delivers a signal at the END output 42 on the rear panel. The polarity of the end signal can be selected with the POLARITY switches 37 on the rear panel. In case of positive polarity, the END output delivers high-level signals during data output, otherwise low-level signals, e.g. during a pause. Also, one of these switches can be used to select whether the stop input is to respond to the positive-going or the negative-going edge.

2.2.1.3.4 Adjusting the DAV Signal

After the data appear at the output of the data drivers, the IGA waits for a moment and then feeds the DAV signal to the clock driver. This wait time and the width of the DAV signal can be set with two potentiometers 36 on the rear panel. With the potentiometers turned to the left-hand stop, reliable operation is always ensured since, in this case, both the wait time and the width are small. However, if the IGA is operated with a high output frequency and

at the same time a long wait time and a large pulse width are selected, it may happen that the DAV signal timing is no longer suitable for the data output clock. Therefore it is recommended that the setting of the wait time and of the width be checked on an oscilloscope:

Output high and low alternately on a data channel.
Monitor DAV signal on one channel of the oscilloscope and data flow on the other channel.



2.2.1.3.5 Single-stepping

STEP ↓ STEP ↑

The keys STEP ↓ 16 and STEP ↑ 16 (SHIFT) permit line-by-line data output; the line being output is displayed.

2.2.1.3.6 Clock Modes

CLOCK HANDSHAKE MODE

The CLOCK MODE key 18 (SHIFT FREQ) permits the selection of data output with internal clock, external clock (input on the rear panel), or two-wire or three-wire handshake in accordance with IEC 625. The control lines DAV, NRFD, NDAC are used for the three-wire handshake and the control lines DAV and NDAC for the two-wire handshake (for the handshake timing see 2.6 and 2.7 as well as Figs. 2-6 and 2.7).

2.2.1.3.7 Output Modes

CYCLE SINGLE REPEAT MODE

The CYCLE MODE key 20 permits the selection of single-shot or repetitive output of the data set. The POLARITY LOGIC switch 37 on the rear panel also enables negative logic to be set, i.e. a binary 1 on the display corresponds to a low signal at the data driver output.

2.2.2 Modules Defined by a Name

All the operating instructions for the direct module are also applicable for modules defined by a name.

2.2.2.1 Assigning a Module Name

MODULE NEW

The name of a module must be keyed in prior to the data entry. The data are managed under this name like the files in a computer. Data cannot be entered into the IGA without giving them a name (except for the direct module).
The name consists of a maximum of six alphanumeric characters. It is possible to use all the numbers and letters, and the space character; however, the latter should not be the first character of a name. It is not possible to use the same name for two modules. The following example shows how a new module defined by the name "ALFA 1" is created.

Key pressed			Display on the IGA
		NEW	
SHIFT	MODULE	OLD	?
		A	A _
		L	
SHIFT		9	AL
		F	ALF_
		A	ALFA_
SHIFT		MODE	ALFA_
		1	ALFA 1
		ENTER	ALFA 1

2.2.2.2 Fetching an Existing Module

MODULE | OLD |

Modules previously stored in the IGA can be fetched again. The corresponding module then becomes the current one, i.e. it can be applied to the data drivers with the START key 8. All the entries and displays refer to this current module. The previously current module is automatically stored in the IGA.

The following example shows how a module defined by the name "ALFA 1" is fetched to become the current one:

Key pressed		Display on the IGA
MODULE	OLD	?
	A	A _
	L	
SHIFT	9	AL
	F	ALF_
	A	ALFA_
SHIFT	MODE	ALFA_
	1	ALFA 1
	ENTER	ALFA 1

2.2.2.3 Menu of the Modules Stored in the IGA and Changing of a Module Name

MODULE NAME

The IGA contains a menu of the modules stored. This menu can be read out on the left-hand side of the display. For this purpose, the NAME key 31 is pressed in the MODULE section. After pressing this key, the name of the current module appears on the display. The remaining elements of the menu can be displayed with the aid of the keys ↑ 29 and ↓ 29a. The module whose name is being displayed automatically becomes the current one and can be applied to the data drivers with the START key 8. As long as the name is being displayed and the cursor is blinking, the name can be changed by overwriting. The new name replaces the old one as soon as the ENTER key 15 has been pressed.

Example: The IGA contains the modules DIRECT, M1, M2, M3

Key pressed		Display on the IGA	Remarks
Module	NAME	DIRECT	The module DIRECT is the current one
	↓	M1	Module M1 is the current one
	↓	M2	Module M2 is the current one
	↓	M3	Module M3 is the current one
	A	A3	
	2	A2_	
	ENTER	A2	The name of module M3 has been changed to A2

2.2.2.4 Erasing of Modules

KILL

The module to be erased is displayed with the aid of the NAME key 31 and the keys ↑ 29 and ↓ 29a. The erasure is initiated with the KILL key 21 (SHIFT). The display signals KILL??. If the KILL key has been pressed by mistake, this function can be cancelled by pressing the CLEAR key 29a. The module is not actually erased until the ENTER key 15 is pressed.

Key pressed		Display on the IGA	Remarks
Module	NAME	M1	Module M1 is the current one
	↓	M2	Module M2 is the current one
	KILL		
SHIFT	→	KILL??	
	ENTER	M1	Module M2 is erased Module M1 is the current one

2.2.2.5 Duplication of Modules

DUPL

In many cases, an already-entered module is to be changed; however, it may be desired to retain a copy of the old module. To this effect, the module is duplicated under a new name. The name of the required module is displayed with the aid of the NAME key 31 and the cursor keys 29. Then the DUPL key 22 (SHIFT ←) is pressed, the new name is entered, and then it is confirmed with the ENTER key 15. Then the IGA stores a duplicate of the old module under the new name. In this way, the contents of the direct module can also be retained in the IGA under a different name.

Key pressed		Display on the IGA	Remarks
Module	NAME	M1	Module M1 is the current one
	DUPL		
SHIFT	←	?	
	A	A_	The name of the duplicate is entered
	0	A0_	
	ENTER	READY	A duplicate of module M1 is stored under the name A0

2.2.3 Sequences

If the IGA is required to output a number of modules one after the other, these modules are combined to form a sequence. The order in which the modules are to be output is determined in a list (see section 2.2.3.3).

All the modules of a sequence must use the same number of channels.

2.2.3.1 Assigning a Module Sequence Name

MODULE SEQ | NEW

The name of a module sequence has to be keyed in prior to entering the list of module names which are to make up the sequence. Without a name, elements of a module sequence cannot be entered. The same restrictions apply for the name of a module sequence as for that of a module. The direct module cannot be part of a sequence. The following example shows how a module sequence defined by the name "BETA 5" is created.

Key pressed			Display on the IGA
		NEW	
SHIFT	MODULE SEQ	OLD	?
		B	B _
		E	BE_
		T	
SHIFT		1	BET_
		A	BETA_
SHIFT		MODE	BETA _
		5	BETA 5
		ENTER	BETA 5

2.2.3.2 Fetching an Existing Module Sequence

MODULE SEQ | OLD

Module sequences which the user has previously stored in the IGA can be fetched again like modules. The corresponding module sequence then becomes the current one, i.e. it can be applied to the data drivers with the START key 8. All the entries and displays then refer to this current module sequence. Any pauses specified in the modules have an effect in the sequence only if they are longer than 0.4 s. If a sequence includes one or more pauses of > 0.4 s and to be repeated, a pause of 1 s is automatically added at the end of the sequence. If no specific frequency is entered for the sequence, the output is performed at the basic frequency setting of 10 kHz. The frequency, end tristate, cycle and clock modes entered for the individual modules do not affect the run of the sequence. The previously current module or module sequence is automatically stored. As soon as a module sequence has become the current one, new elements can be added to the sequence or old elements can be cancelled; however, it is not possible to change the contents of the individual modules. For this purpose, the corresponding module must be fetched to become the current one.

To make a module sequence the current one, first press the MODULE SEQ OLD key 3, then enter the name of the module sequence and confirm with the ENTER key 15.

Example: Causing sequence S1 to become the current one

Key pressed		Display on the IGA
MODULE SEQ	OLD	?
	S	
SHIFT	0	S _
	1	S1_
	ENTER	S1

2.2.3.3 Entering Module Names to Create a Module Sequence

SEQ

First, press the MODULE SEQ NEW key 3 to update a module sequence. Then the SEQ key 4 initiates the entry of module names. A 1 is displayed indicating that the first module of the sequence can be entered. Next the module name appearing on the left-hand side of the display is entered. The entry is confirmed with the ENTER key 15. Then a 2 appears and the second module name can be entered, etc. As long as the cursor bar is blinking, the entries can be changed by overwriting. The keys ↑ 29 and ↓ 29a permit lines which have already been confirmed by the ENTER key to be displayed again and changed.

Key pressed		Display on the IGA		Remarks
	NEW			
SHIFT MODULE SEQ	OLD	?		
	S			
SHIFT	0	S_		The module sequence S1 is created
	1	S1_		
	ENTER	S1		
	SEQ	?	Ø1	
	A	A_	Ø1	Module A is the first module of the sequence
	ENTER	?	Ø2	
	B	B_	Ø2	Module B is the second module of the sequence
	ENTER	?	Ø3	
	C	C_	Ø3	Module C is the third module of the sequence
	ENTER	?	Ø4	

A module forming part of a sequence can be erased with the aid of the KILL key. The module name is displayed as described above and then the KILL key 21 is pressed. Then pressing the ENTER key causes the actual erasure.

Example: Erasing module B from a sequence

Key pressed		Display on the IGA		Remarks
MODULE SEQ	OLD	?		
	S			
SHIFT	0	S_		
	1	S1_		Module sequence S1 is made the current one
	ENTER	S1		
	SEQ	A	Ø1	The first module of the sequence is displayed
	↓	B	Ø2	The second module of the sequence is displayed
	KILL			
SHIFT	→	KILL??	Ø2	Module B is erased from the sequence
	ENTER	C	Ø2	The next module of the sequence is displayed

2.2.3.4 Menu of the Module Sequences Stored in the IGA; Erasing Module Sequences

MODULE SEQ | NAME

As for the modules, the IGA contains a menu of the module sequences stored. This menu is read out in the same way as for the modules, except that, in this case, the NAME key 2 in the module sequence section must be pressed.

The module sequences are erased in the same way as the modules; however, for this purpose, the NAME key in the module sequence section must be used again.

Key pressed		Display on the IGA	Remarks
MODULE SEQ	NAME	S1	Sequence S1 is the current one
	↓	S2	Sequence S2 is the current one
	↓	SA	Sequence SA is the current one
	↓	B7	Sequence B7 is the current one
	C	C7	
	O	CO	The name of sequence B7 is changed to read CO
	↑	SA	Sequence SA is the current one
	KILL		
SHIFT	→	KILL??	
	ENTER		Sequence SA is erased

2.3 Cassette Drive (Option IGA-B2)

The built-in cassette recorder permits modules and module sequences to be stored on magnetic tape in order to establish, for instance, a data library or to read the data later into a second IGA. The tape can be used on both sides; however, the following details apply only for one side.

2.3.1 Tape Formatting

CALL · MARK

If a tape side is to be recorded on for the first time, it must first be formatted. For this purpose, the tape needs to be divided into 7 sections each of which is able to accommodate at the maximum the complete contents of the IGA. First, the tape is inserted into the recorder and the cover closed. Next, the formatting is started by the following keystrokes:

Key pressed		Display on the IGA
	CALL	?
	.	
SHIFT	PAUSE	._
	M	
SHIFT	A	.M_
	A	.MA_
	R	
SHIFT	7	.MAR_
	K	
SHIFT	8	.MARK_
	ENTER	.MARK

After the whole of the tape side has been formatted, the IGA signals READY.

2.3.2 Storing Data on Tape

SAVE

Either the complete contents of the IGA, individual modules or module sequences can be written on the tape. In each case, the section to be used for storing has to be selected. Each tape section is able to store the complete contents of the IGA, a module or a module sequence. Data previously stored in the section indicated are overwritten. When storing modules or module sequences, the name of the module or module sequence has to be entered in addition to the section number.

Note: Although the direct module is stored on tape, it can be loaded again only with the aid of the instruction SUBSTITUTE (see section 2.3.3.2). It is best to change the name of the direct module prior to storing (see section 2.2.2.3).

Examples:

Storing the contents of the IGA in section 5:

Key pressed		Display on the IGA	
	SAVE		
SHIFT	LOAD		-
	5	?	5
	ENTER		5
		SAVE	when section 5 is reached
		READY	when read-in is finished

Storing the module ALFA in section 3

Key pressed		Display on the IGA	
	SAVE		
SHIFT	LOAD		_
	3	?_	3
	A	A_	3
	L		
SHIFT	9	AL_	3
	F	ALF_	3
	A	ALFA_	3
	ENTER	ALFA	3

2.3.3 Loading Data from Tape

LOAD .A

2.3.3.1 APPEND Instruction

The load instruction APPEND transfers data from the tape into the IGA without affecting the data contained in the IGA. If the IGA does not have enough space to accommodate the data from the tape or if a data set is already stored under the same name, an error message is output and the load instruction is not executed.

Example:

Loading the contents of section 4 into the IGA:

Key pressed		Display on the IGA		
	LOAD		_	
	4	?	4	
	.			
SHIFT	PAUSE	.	4	
	A	.A_	4	
	ENTER	.A	4	
		LOAD		as soon as section 4 is reached
		READY		when loading is finished

Note: A direct module stored on the tape is not loaded by the APPEND instruction. The module can be loaded by the LOAD.S instruction.

2.3.3.2 SUBSTITUTE Instruction

LOAD .S

The load instruction SUBSTITUTE transfers data from the tape into the IGA, including the direct module; simultaneously all the old data in the IGA are erased.

Example:

Loading the contents of section 5 into the IGA

Key pressed		Display on the IGA	
	LOAD		-
	5	?	5
	.		
SHIFT	PAUSE	.	5
	S		
SHIFT	Ø	.S_	5
	ENTER	.S	5
		LOAD	as soon as section 5 is reached
		READY	when loading is finished

2.3.4 Write Protection

The tape can be write-protected by removing the plastic plugs from the cassette (see Fig. 2-5).

2.3.5 Erasing the Complete Contents of the IGA

CALL . DEL

Loading especially large data files which require almost the entire storage capacity virtually always causes an error message to be issued, unless previously all the modules stored in the IGA have been erased. To simplify this procedure, a special instruction is available for erasing the entire IGA contents. For this purpose, the following keys have to be pressed:

Key pressed		Display on the IGA
	CALL	?
	.	
SHIFT	PAUSE	._
	D	D_
	E	.DE_
	L	
SHIFT	9	.DEL_
	ENTER	.DEL??

The contents are erased only after the ENTER key has been pressed again.

ENTER

2.4 Controlling the IGA via the IEC Bus (Option B1)

The IGA is fitted with a remote-control interface in accordance with IEC 625-1. The IEC-bus connector 44 is located on the rear panel. Fig. 2-1 shows the pin allocation:

Fig. 2-1 Pin allocation

The only characters used for data transfer are the conventional ASCII characters (table 2-3). For the interface specifications (control lines, handshake lines, data lines) and the timing of the data transfer see the IEC standard.

2.4.1 Setting the Device Address and the Terminating Character

The listen and the talk addresses are set together by the coding switches 44 S1/A1 to A5 in accordance with table 2-1. The IGA is delivered with the listener address "," and the talker address "L" factory-set corresponding to device address 12 (for use with e.g. the R&S Process Controller PPC).

The CR/LF switch 44 can be used to select the terminating character; either CR (carriage return) or LF (line feed).

Table 2-1 Setting the device address

ASCII character		Binary address					
Listen address	Talk address	Address switch A5	A4	A3	A2	A1	Decimal equivalent
(SPACE)	@	Ø	Ø	Ø	Ø	Ø	Ø
!	A	Ø	Ø	Ø	Ø	1	1
"	B	Ø	Ø	Ø	1	Ø	2
	C	Ø	Ø	Ø	1	1	3
$	D	Ø	Ø	1	Ø	Ø	4
%	E	Ø	Ø	1	Ø	1	5
&	F	Ø	Ø	1	1	Ø	6
'	G	Ø	Ø	1	1	1	7
(	H	Ø	1	Ø	Ø	Ø	8
)	I	Ø	1	Ø	Ø	1	9
*	J	Ø	1	Ø	1	Ø	1Ø
+	K	Ø	1	Ø	1	1	11
,Comma	L	Ø	1	1	Ø	Ø	12*
-	M	Ø	1	1	Ø	1	13
.	N	Ø	1	1	1	Ø	14
/	O	Ø	1	1	1	1	15
Ø	P	1	Ø	Ø	Ø	Ø	16
1	Q	1	Ø	Ø	Ø	1	17
2	R	1	Ø	Ø	1	Ø	18
3	S	1	Ø	Ø	1	1	19
4	T	1	Ø	1	Ø	Ø	2Ø
5	U	1	Ø	1	Ø	1	21
6	V	1	Ø	1	1	Ø	22
7	W	1	Ø	1	1	1	23
8	X	1	1	Ø	Ø	Ø	24
9	Y	1	1	Ø	Ø	1	25
:	Z	1	1	Ø	1	Ø	26
;	[	1	1	Ø	1	1	27
<	\	1	1	1	Ø	Ø	28
=	]	1	1	1	Ø	1	29
>	^	1	1	1	1	Ø	3Ø

*) factory-set

2.4.2 Interface Functions

The following interface functions are realized on the IGA:

SH1	Source handshake, full capability
AH1	Acceptor handshake, full capability
T5	Talker, capability to reply to serial poll, unaddressing if MLA, talk-only mode
L4	Listener, unaddressing if MTA
SR1	Service request, full capability
RL1	Remote/local, full capability
DC1	Device clear, full capability
DT1	Device trigger, full capability

2.4.3 Setting Instructions

The setting instructions for remote control of the front-panel keys and for functions which cannot be selected manually are described below. The IGA responds to IEC-bus setting instructions and data in the remote mode in the same way as to direct manual entries in the local mode. Therefore the same sequence of settings can be used when writing the controller program as for manual operation.

The programming instructions comprise a header consisting of two characters, the subsequent data, if any, and a delimiter.

Transmitting the header corresponds to pressing a command key (see table 2-3) which either causes an operation immediately (e.g. start, stop, cycle mode, clock mode, etc.) or makes the IGA ready for subsequent data entry (e.g. data, format, frequency, etc.). Keys performing a switchover function (clock mode, cycle mode) are assigned a special instruction for each of the modes which they can select.

For data coding see table 2-2.

Each instruction must be terminated by the delimiter ";".
The ";" follows either directly after the header if the programmed key itself causes an action, or after the last datum if one or several data are listed after the header.

Example:

↑ "SP;MOALFA;CØ;FR1E3;ST;"

SP stops the data output, MO updates the old module ALFA, CØ sets CLOCK INTERN and FR a frequency of 1 kHz, and ST starts data output (the headers are underlined in the examples given).

If several, similar data follow the header, the data must be separated by the "," character.

The different delimiters "," and ";" are necessary to obtain clear remote-control instructions. The letter sequence DA can either stand for the DATA key or the hexadecimal numbers D and A in case of data input.

The IGA assumes that all characters represent function keys (e.g. DATA key). Only when such an instruction has been recognized switches the IGA to the data reception mode which allows to evaluate the characters as hexadecimal numbers. The data reception mode is not abandoned until the ";" has been received. The delimiter "," does not terminate the data reception mode.

Example:

↑ "DA8FE37E65,76349EDA,3FE1Ø3BC;"

This instruction enters three consecutive data lines, beginning with the first data line of the module.

"LI 3Ø;"	Data line 3Ø is displayed
"LI 3Ø, ABC8, 6543;"	Data line 3Ø is filled with the data ABC8 and the following line with 6543
"LI 4Ø, #;"	Data line 4Ø is erased
"LI 4Ø, S2Ø;"	The contents of data line 2Ø are written into data line 4Ø

"FO28H4B,ABCD;"

This instruction selects a format of 4 lines for binary entry and 28 lines for hexadecimal entry, and selects the end tristate mode for the data drivers A, B, C and D.

"FO14B8H,;"

This instruction selects a format of 14 lines for binary entry and of 8 lines for hexadecimal entry.

"CA.DEL,;"	The complete IGA contents (Call. DEL) are erased
"LO1.S;"	Section 1 is deleted from the tape as soon as the IGA contents have been erased (Load .S)
"SA2 DEMO;"	The module or the module sequence named DEMO is stored in section 2 on the tape

The following keys cannot be remote-controlled:

DEL, ↑ , ↓ , ← , → , ENTER

The function of the ENTER key is performed by the delimiters ";" and ",".

In addition to remote control of the front-panel keys, other modes which are possible only with the IEC bus can be programmed.

2.4.4 Data Direct Instruction

The DD (data direct) instruction permits data to be applied directly to the data drivers via the IEC bus.

Examples:

The instruction IECOUT 12;"DDAF45732E;" applies the following datum to the individual data drivers:

Driver A	2E
Driver B	73
Driver C	45
Driver D	AF

The instruction IECOUT 12;"DDAF45..2E;" applies the following datum to the individual data drivers:

Driver A	2E
Driver B	Tristate
Driver C	45
Driver D	AF

If more than 8 characters are sent after the letters DD, only the last 8 characters before the semicolon are taken into account.

2.4.5 Printer Output

CALL . PRINT

To facilitate logging the data stored in the IGA, the Logic Generator is able to output, via the IEC-bus interface, the data of the current module or the listing of the individual module names of the current sequence on an IEC-bus-compatible printer. The data are printed out with the line number in the same way as they appear on the display. The printer operates in the listen-only mode and the IGA in the talk-only mode.

Pressing the following keys causes the printout:

		Display on the IGA
	CALL	?
	.	
SHIFT	PAUSE	._
	P	
SHIFT	5	.P_
	R	
SHIFT	7	.PR_
	I	
SHIFT	E	.PRI_
	N	
SHIFT	B	.PRIN_
	T	
SHIFT	1	.PRINT_
	ENTER	.PRINT

After the last character has been output, the IGA disables the talk-only mode and signals READY.

If no printer with listen-only mode is available, the same text can also be read into the bus controller with the aid of the IEC-bus instruction "DO;".

Example: A module with 10 data lines is to be read out and output on the display of the bus controller.

```
100  IECOUT 12, "DO;"      Data output is prepared
110  FOR I = 1 TO 10
120  IECIN 12, A$          A data line is read out
130  PRINT A$              A data line is output on the display
140  NEXT I
```

Table 2-2

IEC-bus character	Meaning of key
Ø to 9	Ø to 9
A	A
B	B, BIN
C	C
D	D, DEC
E	E
F	F
G	G
H	H, HEX
I	I
J	J
K	K
L	L
M	M
N	N
O	O, OCT
P	P
Q	Q
R	R
S	S
T	T
U	U
V	V
W	W
X	X
Y	Y
Z	Z
.	.
	(SPACE)
#	KILL
S	DUPL

Table 2-3

IEC-bus characters	Meaning of key	
MO	MODULE OLD	
MN	MODULE NEW	
SO	MODULE SEQ OLD	
SN	MODULE SEQ NEW	
DA	DATA	
SS	SEQ	
NM '	MODULE NAME '	
NS	SEQ NAME	
FR	FREQ	
PA	PAUSE	
LI	LINE	
FO	FORMAT	
RE	RENUM	
CA	CALL	
LO	LOAD	
SA	SAVE	
ST '	START	
SP '	STOP	
SU '	STEP	
SD '	STEP	
CX	CLEAR	
MR '	CYCLE MODE REPEAT	direct instructions
MS '	CYCLE MODE SINGLE	
CØ	CLOCK INTERNAL	
C1	CLOCK EXTERNAL	
C2	2-WIRE-HANDSHAKE	
C3	3-WIRE-HANDSHAKE	

2.4.6 Group of Addressed and Universal Instructions

2.4.6.1 Remote/Local

As soon as the IGA receives its listen address from a controller, it switches to the REMOTE mode and remains in this status even after termination of the data transfer. In the REMOTE mode, which is indicated by the LED 13, all the front-panel keys are disabled. If the IGA receives the instruction GTL (go to local) or if the LOCAL key 14 is pressed, the instrument switches to LOCAL, i.e. manual setup is now possible again.

The LOCAL key can be inhibited by the controller sending the instruction LLO (local lockout).

2.4.6.2 Device Clear

The universal instruction DCL (device clear) or the addressed instruction SDC (selected device clear) produce the same effect as the CLEAR key.

2.4.6.3 Device Trigger

As soon as the IGA receives the GET instruction (group execute trigger), it starts outputting the current data. This trigger instruction corresponds to the IEC-bus instruction ST.

Table 2-4 gives examples of the output obtained with the instructions described above; address 12 is assumed to be the IEC-bus address of the IGA.

Table 2-4 General IEC-bus instructions

2.4.6.4 Service Request, Status Byte, Serial Poll

If the operator makes a mistake causing an error message to be displayed on the IGA, the instrument sets the SRQ line to LOW.

A serial poll can then be carried out to determine whether the IGA has issued the service request. Next a status byte can be read out, its decimal equivalent corresponding to the error number displayed (see section 2.5).

The following example shows how the Process Controller PPC performs a serial poll after a service request message from the IGA and how the status byte is read out.

Main program:

```
100       IECSRQ GOTO 3000
          Serial poll routine:
3000      IECSPE
3010      IEC TAD12
3020      IEC%IN U%                    U% contains the status byte
3030      IECUNT
3040      IECSPD
3050      IEC RET SRQ
```

Example: U% = 65 This means that the data memory of the IGA is full.

Table 2-5 ASCII code

2.5 Error Messages

The display of the error message can be cleared by pressing the CLEAR key.

Error No.	Type of error	Remarks
65	Data memory full	
66	Comments memory full	Can be remedied by renumbering modules
67	Format as entered is invalid	More than 32 channels or the number entered is not a multiple of 3 in the case of octal entries or of 4 in the case of hex entries
68	Name already used	
69	Number entered too high or too low	
70	Line number too high	Maximum 1Ø24Ø, see section 2.2.1.2.4
71	Module not found	
72	Module sequence not found	
73	Entry does not correspond to format specified	e.g. 8 in the case of octal or 3 in the case of binary entries
74	CMOS RAM data lost	Check charge state of battery, and charging circuitry
75	Magnetic tape write-protected	Reinsert plastic plugs into cassette (see Fig. 2-5)
76	No tape inserted	
77	Tape option not fitted	
78	Write error on the tape	
79	Rear error from the tape	
80	IEC-bus option not fitted	
81	Syntax error	
82	The current module or all the modules of the current module sequence do not contain any data or the current module sequence does not contain any module.	

Error No.	Type of error	Remarks
83	No pause < 0.4 s possible since at least one module of the sequence has a pause of ≥ 0.4 s	See section 2.2.3.3
84	The module name called up is the name of a module sequence or vice versa.	
85	The module name entered has previously been used in this sequence.	
86	Single stepping is not possible in the handshake mode.	
87	The direct module must not be included into a module sequence.	
88	One module of the sequence is not found.	
89	Line tristate or DAV tristate not possible since the format channels required for line tristate or DAV tristate are occupied.	See section 2.2.1.2.10
90	The format specifications for the individual modules of the sequence do not match.	One module defines tristate, in another one, for instance, channel 32 is used for data.
91	Data line not existing	
92	Module cannot be executed as program	
93	The direct module cannot be recorded on the magnetic tape as individual module	Can be stored only together with the entire IGA contents
94	The validity range of the tristate points does not coincide with the limits of the data drivers	e.g. two bits of a hex number contained in data driver A, the other 2 bits in data driver B
96	Error of IEC-bus syntax	In most cases wrong use of delimiters , ;
97	Error of IEC-bus handshake	e.g. no IEC-bus cable connected

2.6 List of Events for 3-wire Handshake Operation

(See Fig. 2-6)

1 At the beginning, the IGA is holding the DAV signal high (data not valid).

2 At the beginning, the items under test are holding the NRFD (none is ready to accept data) and the NDAC (none has accepted data) signals low.

3 The IGA now applies the data byte to the data lines.

4 The IGA waits to allow the data lines to settle.

5 All the test items indicate their readiness to accept the first data byte; the NRFD line is brought high.

6 After verification that the NRFD signal is high, the IGA brings the DAV signal low to indicate that the data lines have settled and the data are valid.

7 The first test item brings the NRFD signal low to indicate that it is no longer in the ready state, then it accepts the data. The remaining test items follow after their own processing times.

8 The first test item brings the NDAC signal high to indicate that it has accepted the data (The NDAC signal remains low as long as there are other test items which keep the NDAC signal low).

9 The last test item brings the NDAC signal high to indicate that it has accepted the data; all the test items have now accepted the data and the NDAC line goes high.

10 After the IGA has recognized that the NDAC signal has been brought high, it brings the DAV signal high. This signals to the test items that the data on the data lines have to be regarded as not valid.

11 The IGA changes the data on the DIO lines.

12 The IGA waits to allow the data lines to settle.

13 After recognizing that the DAV signal is high (at 10), the test items bring the NDAC signal low in order to initialize the next cycle. The NDAC line is brought low as soon as the first test item has brought this signal low.

14 By bringing the NRFD signal high, the first test item indicates that it is ready to accept the next data byte (The NRFD signal remains low as long as other test items keep the NRFD signal low).

15 By bringing the NRFD signal high, the final test item indicates that it is ready to accept the next data byte; then the NRFD line goes high.

16 After the IGA has recognized that the NRFD signal has been brought high, the IGA brings the DAV signal low to indicate that the data lines have settled and the data are valid.

17 The first test item brings the NRFD signal low to indicate that it is no longer in the ready state, then it accepts the data.

18 The first test item brings the NDAC signal high to indicate that it has accepted the data (same as under 8).

19 The last test item brings the NDAC signal high to indicate that it has accepted the data (same as under 9).

20 After the IGA has recognized that the NDAC signal has been brought high, it brings the DAV signal high (same as under 10).

21 After bringing the DAV signal high, the IGA changes the data byte on the data signal lines.

22 After recognizing that the DAV signal is high, the test item bring the NDAC signal low to initialize the next cycle.

23 All the three handshake lines are now back in their initial state as under 1 and 2.

2.7 List of Events for 2-wire Handshake Operation

(See Fig. 2-7)

1 At the beginning, the IGA is holding the DAV signal high (data not valid).

2 At the beginning, the test item is holding the NDAC signal low (the test item has not yet accepted the data).

3 The IGA now applies the data byte to the data lines.

4 The IGA waits to allow the data lines to settle. Then the IGA brings the DAV signal low to indicate that the data lines have settled and the data are valid.

5 The test item brings the NDAC signal high to indicate that it has accepted the data.

6 After the IGA has recognized that the NDAC signal has been brought high, it brings the DAV signal high. This signals to the test item that the data on the data lines now have to be regarded as not valid.

7 The IGA changes the data on the DIO lines.

8 After recognizing that the DAV signal is high (at 6), the test item brings the NDAC signal low in order to initialize the next cycle.

9 The IGA waits to allow the data lines to settle.
After recognizing that the NDAC signal is low, the IGA brings the DAV signal low to indicate that the data lines have settled and the data are valid.

10 The test item brings the NDAC signal high to indicate that it has accepted the data (same as under 5).

11 After the IGA has recognized that the NDAC signal is high, it brings the DAV signal high (same as under 6).

12 After bringing the DAV signal high, the IGA changes the data byte on the data lines.

13 After recognizing that the DAV signal is high, the test item brings the NDAC signal low in order to initialize the next cycle.

14 All the handshake lines are now back in their initial state as under 1 and 2.

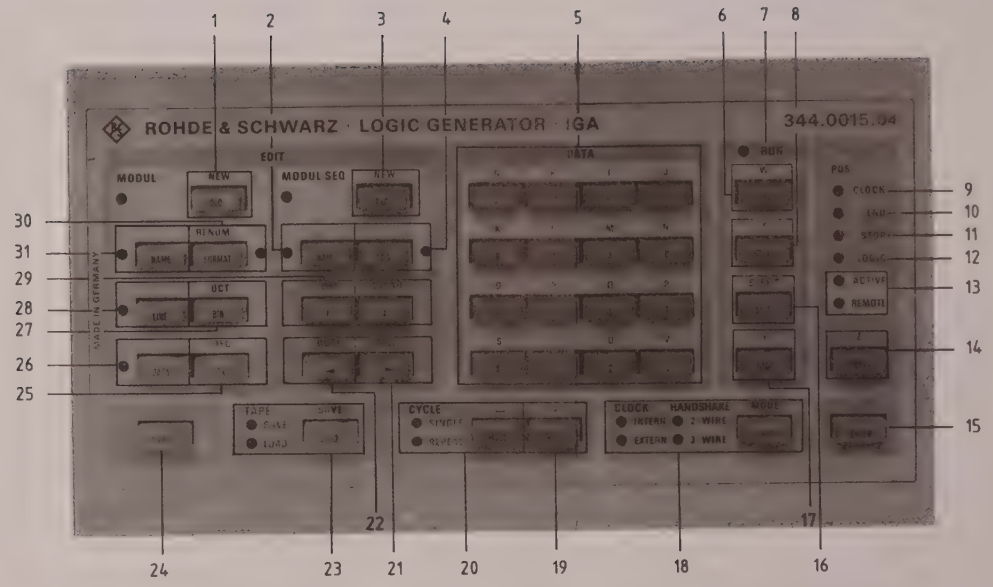


Bild 2-2 Frontansicht
Fig. 2-2 Front view

344.2760 - 2.63

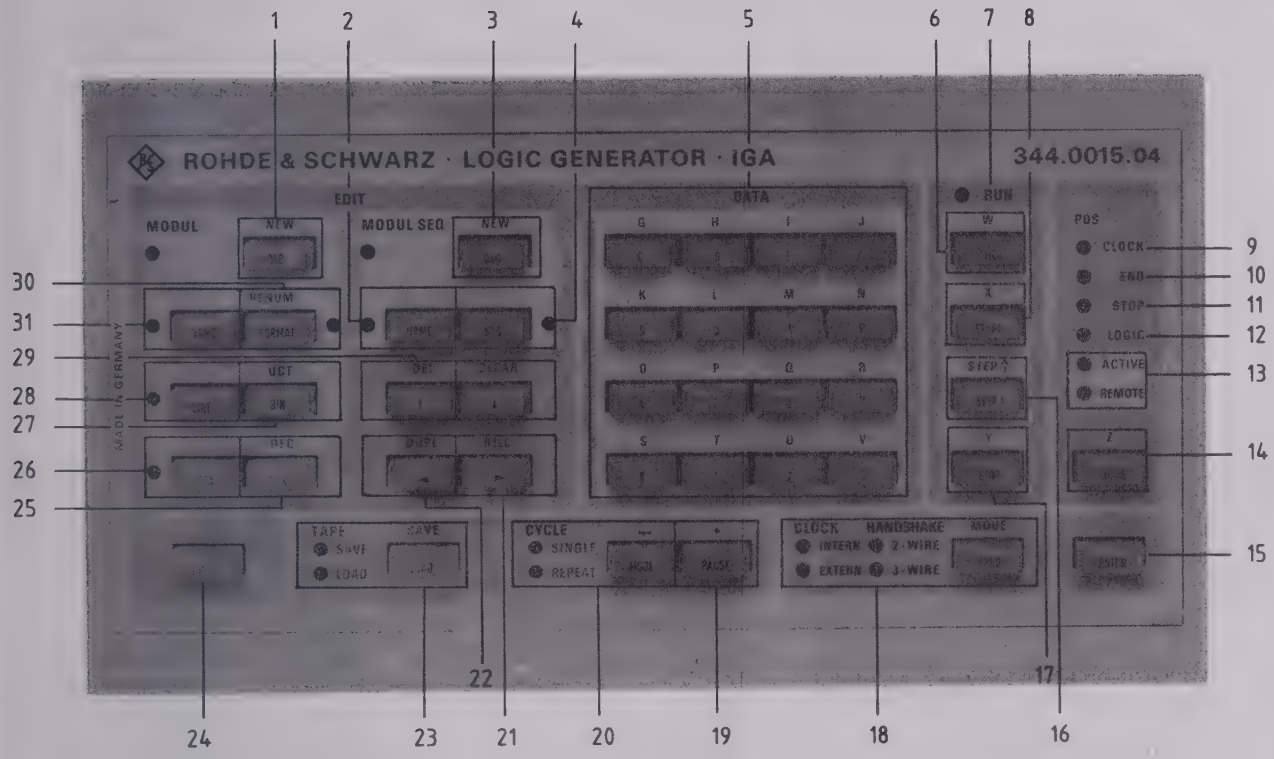


Bild 2-2 Frontansicht
Fig. 2-2 Front view

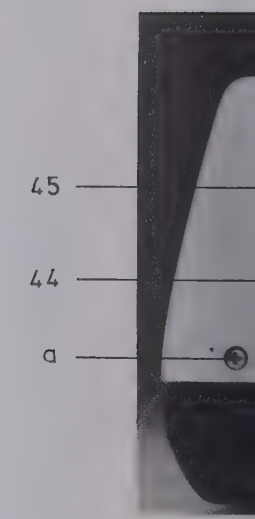
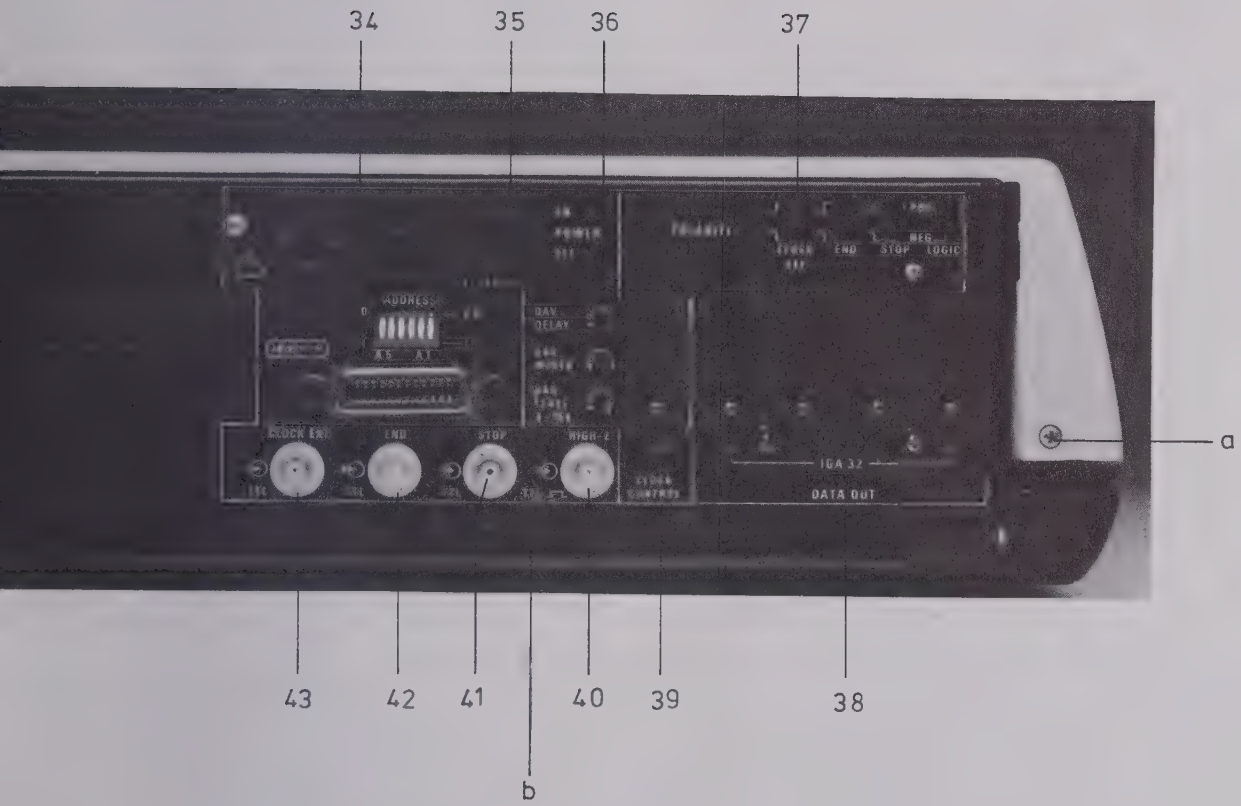


Bild 2-3 Rü
Fig. 2-3 Re


34
35
36
37
a
43
42
41
40
39
38
b
ADDRESS
DAC DELAY
END
STOP
LOGIC
DATA OUT

nsicht
view

344.2760-2.65

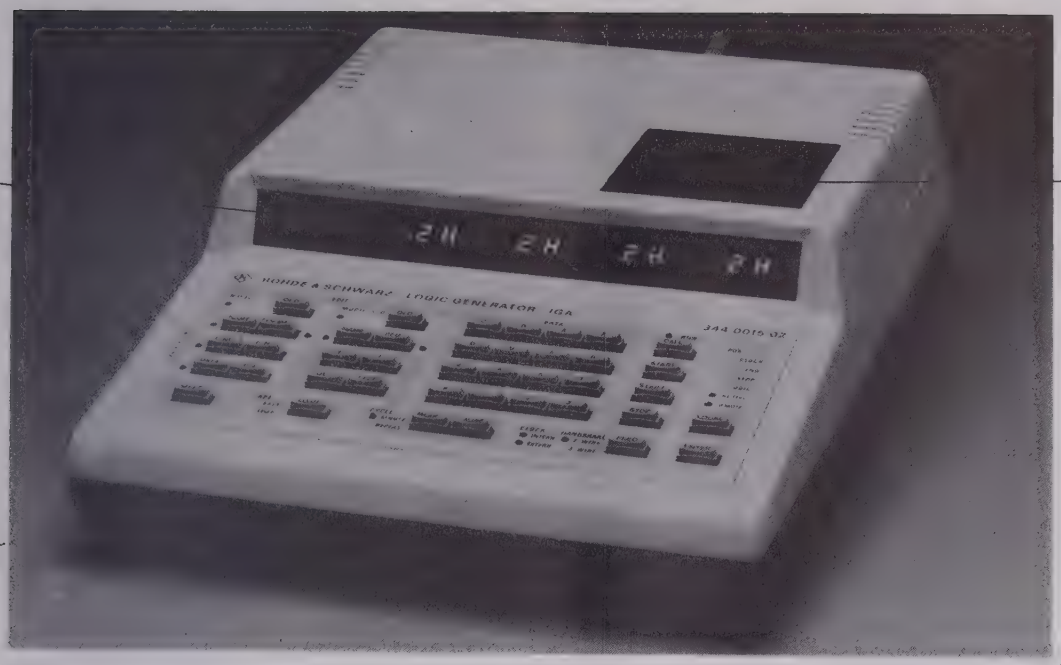


Bild 2-4 Logic Generator IGA
Fig. 2-4 Logic generator IGA

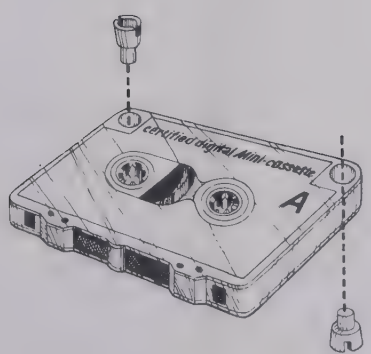
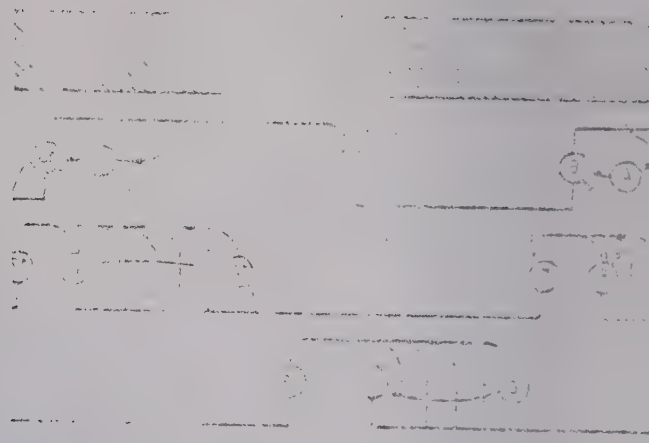
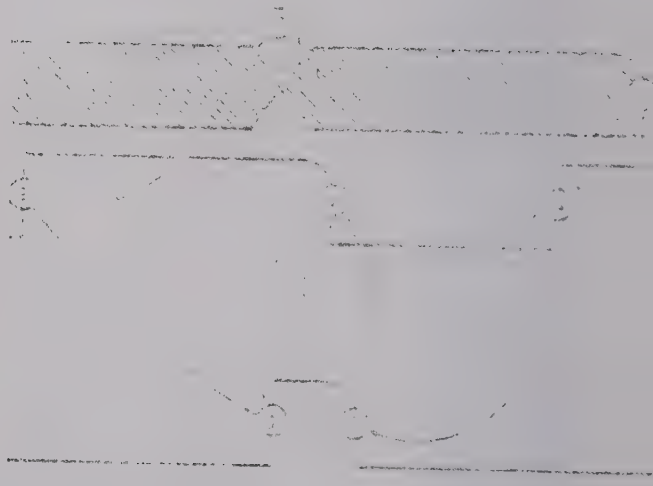
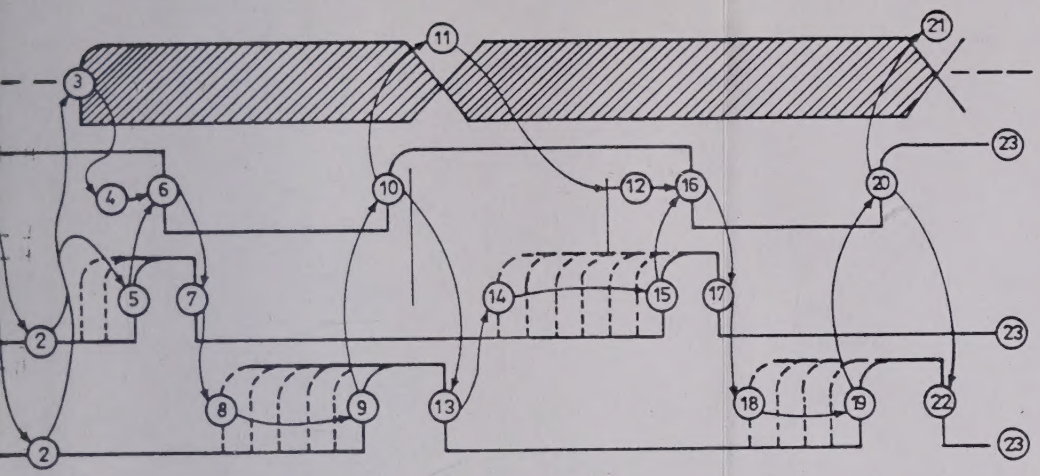


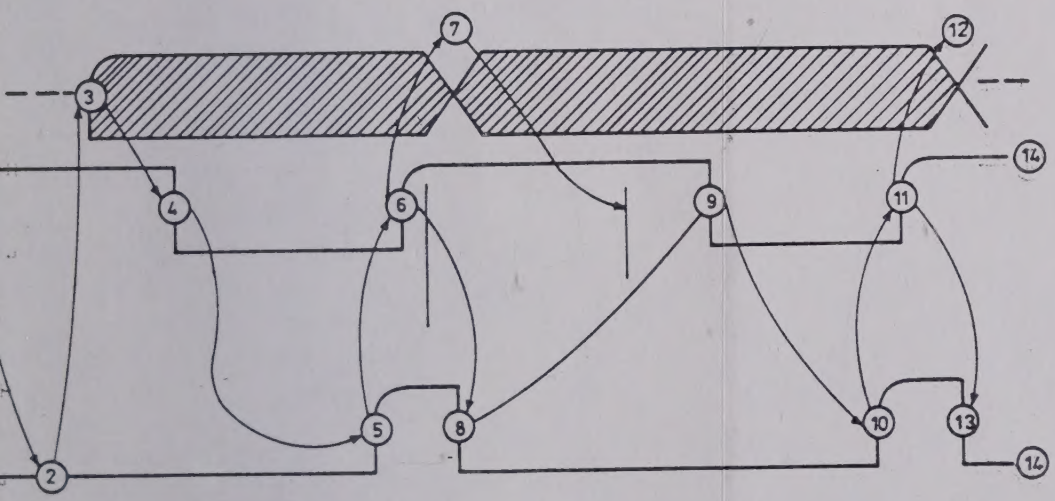
Bild 2-5 Entfernen des Schreibschutzes
Fig. 2-5 Removing the write protection







2-6 Diagramm des zeitlichen Ablaufs des 3-Wire- Handshake
2-6 Timing diagram of 3-wire handshake



2-7 Diagramm des zeitlichen Ablaufs des 2-Wire-Handshake
2-7 Timing diagram of 2-wire handshake

www.manualsplus.com
1003675